UNIVERSIDAD COMPLUTENSE DE MADRID

Proyecto de Innovación

Convocatoria 2019/2020

Nº de proyecto: 139

Herramienta integral y de bajo coste para el desarrollo de prácticas remotas para las asignaturas de Ciencias e Ingeniería

Responsable del proyecto:

Eva Besada Portas

Facultad de Ciencias Físicas

Departamento de Arquitectura de Computadores y Autom´atica

1. Objetivos propuestos en la presentación del proyecto

El objetivo principal de este proyecto es el desarrollo de una herramienta integral y de bajo coste para dar acceso remoto, a través de Internet, a las experiencias de los laboratorios de las áreas de Ciencias e Ingeniería. Como herramienta integral entendemos que ésta debe permitir realizar todas las labores necesarias para poner en marcha las experiencias del laboratorio desde un mismo entorno. Con su bajo coste indicamos que el gasto asociado a conectar un laboratorio existente a internet no sea, salvo por necesidades de sensorización y actuación especiales, superior a los 150 €.

El objetivo general se puede articular en los siguientes dos objetivos diferenciados y en sus correspondientes subconjuntos de propósitos concretos:

Objetivo 1 (O1). Automatizar e integrar en una única herramienta el proceso de generación de los laboratorios remotos de bajo coste de la UCM basados en EJsS, Node.js y la Raspberry PI. Para alcanzarlo, se extenderá las funcionalidad de la herramienta de programación gráfica EJsS con las siguientes capacidades:

O1.A) Facilitar la conexión entre las variables utilizadas en el código del controlador (que interactúa con los sensores y actuadores conectados al dispositivos) y las variables javascript de la página web que da acceso al laboratorio. Es decir, la herramienta EJsS extendida dispondrá de la funcionalidad necesaria para que el usuario pueda conectar de forma automática las variables que el código C envía al servidor con las variables que la página web del laboratorio utiliza para que los alumnos puedan visualizar la evolución del estado del sistema real e interactuar con él.

O1.B) Permitir acceder y editar el código del controlador de la Raspberry PI desde EJsS, y una vez que esté listo, descargarlo y compilarlo desde EJsS sobre la Raspberry PI. Es decir, la herramienta extendida EJsS dispondrá de la funcionalidad necesaria para evitar que el usuario tenga que utilizar herramientas adicionales para modificar y volcar el código más cercano a la experiencia, y lo almacenará como parte del proyecto del laboratorio remoto que ha creado, para evitar la disparidad de versiones entre lo que está volcado en la Raspberry PI y lo que está almacenado en el proyecto de EJsS.

O1.C) Posibilitar el volcado de la página web de HTML5.0 y javascript creada en EJsS por el responsable del laboratorio para que el estudiante pueda interactuar desde su PC, móvil o tableta con la experiencia que debe realizar. Es decir, se eliminará una etapa farragosa del proceso actual, en la que el responsable debe: 1) extraer, del código generado desde EJsS, la parte necesaria para generar la página web de la experiencia; 2) modificarlo ligeramente; y 3) descargarlo sobre los directorios pertinentes de la Raspberry PI.

O1.D) Configurar y descargar la página web de instrucciones del laboratorio remoto. Es decir, el responsable del laboratorio configurará la información necesaria para cada experiencia desde EJsS y ésta será automáticamente descargada sobre la Raspberry PI para que pueda ser utilizada como página web de ayuda dentro del laboratorio remoto.

O1.E) Configurar la información que debe ser almacenada en los ficheros de datos de cada experimento. Es decir, el responsable del laboratorio dejará de tener que modificar el código del servidor Node.js para que éste grabe la información apropiada de cada experimento, utilizando alternativamente para éste fin una nueva funcionalidad que incorporará la versión extendida de EJsS para facilitar y agilizar la puesta en marcha de los laboratorios remotos de bajo coste de la UCM.

O1.F) Simplificar el proceso de alta y baja de usuarios (alumnos) a la página web del laboratorio, por medio de una nueva funcionalidad en EJsS que permita al responsable

del laboratorio dar fácilmente de alta y baja a aquellos alumnos que pueden acceder a la experiencia en diferentes instantes. De esta forma, se agiliza otra etapa del proceso actual, en la que el responsable debe modificar de forma manual una base de datos alojada en la Raspberry PI.

Objetivo 2 (O2). Extender el uso de la herramienta a las experiencias de laboratorio de diferentes asignaturas de Ciencias e Ingeniería. Para lograrlo, se deberán realizar las siguientes labores:

O2.A) Documentar el funcionamiento de la nueva herramienta, para que los responsables del laboratorio dispongan de la información necesaria para crear desde cero un laboratorio remoto. Este propósito resulta fundamental para extender el uso de la herramienta más allá de los profesores involucrados en su desarrollo.

O2.B) Pre-seleccionar y poner a punto los laboratorios remotos de un conjunto experiencias de diferentes campos, teniendo en cuenta las áreas de conocimiento en la que imparten docencia los participantes del proyecto. Más en concreto, se piensa actualizar las prácticas de Control de bajo coste ya disponibles a la nueva metodología, y desarrollar nuevas experiencias de Robótica y Procesamiento de Señales (ambas afines al área de ISA) y de Física del Estado Sólido (afín al área de Materiales).

O2.C) Publicitar y publicar la nueva herramienta que se desarrolle en los foros adecuados. Por una parte, una vez que estén listos los diferentes laboratorios se planea difundir su existencia dentro del ámbito más próximo, el de la propia Facultad de Ciencias Físicas, para explorar la posibilidad de desarrollar nuevas experiencias con esta metodología en un futuro próximo. Por otra, se desea publicar los resultados obtenidos en jornadas, conferencias y revistas relacionadas con el desarrollo de nuevas herramientas educativas para la creación de laboratorios remotos (como podrían ser las Jornadas de Innovación de la UCM, las Jornadas de Automática, expAt o el IEEE Transaction on Education).

En conjunto, los objetivos propuestos se centran en torno a la necesidad de disponer de herramientas integrales que automaticen y agilicen el proceso de puesta a punto de laboratorios remotos a partir de experiencias existentes en laboratorios presenciales, y de extender el uso de este tipo de laboratorios dentro de asignaturas que se puedan beneficiar de su existencia en los campos de Ciencias e Ingeniería, y en nuestro entorno más cercano.

Finalmente, conviene indicar que la herramienta de bajo coste para la creación de laboratorios remotos que se desarrollará en este proyecto resulta especialmente útil en la situación epidemiológica en la que nos encontramos por la COVID-2019, ya que puede permitir que los alumnos realicen desde sus casas, sin tener que desplazarse a sus centros de estudios, parte de las experiencias de los laboratorio presenciales tradicionales.

2. Objetivos alcanzados

El objetivo principal de este proyecto es el desarrollo de una herramienta integral y de bajo coste para dar acceso remoto, a través de Internet, a las experiencias de los laboratorios de las áreas de Ciencias e Ingeniería. Como herramienta integral indicamos que ésta debía permitir realizar todas las labores necesarias para poner en marcha las experiencias del laboratorio desde un mismo entorno, cosa que se ha logrado ya que EJsS se ha convertido en la herramienta de desarrollo y gestión de todo el laboratorio remoto. El requisito de bajo coste de conexión de una experiencia a Internet, salvo para casos especiales de sensorización y actuación especiales, se logra ya que el software (EJsS, Node.js, y controladores programados en lenguajes estándar) que da soporte a la conexión es gratuito y está desplegado sobre una Raspberry PI (que tiene un coste inferior a 110€, cuando se incluye en este valor el de la memoria necesaria para desplegar su software). A continuación analizamos el cumplimiento de los dos objetivos particulares en los que se articula el proyecto.

O1. Centralización del proceso de generación de los laboratorios remotos.

O1.A) *Facilitar la conexión entre las variables utilizadas en el código del controlador (que interactúa con los sensores y actuadores conectados al dispositivos) y las variables javascript de la página web que da acceso al laboratorio.* Este objetivo se ha logrado, ya que el "Plugin" del laboratorio remoto permite que desde EJsS se configuren las variables de intercambio de información a través del servidor e incluye un nuevo Elemento de Modelo que puede ser utilizado por la interfaz gráfica de la experiencia para acceder a la información de estas variables de forma automática.

O1.B) *Permitir acceder y editar el código del controlador de la Raspberry PI desde EJsS, y una vez que esté listo, descargarlo y compilarlo desde EJsS sobre la Raspberry PI.* Este objetivo también se ha logrado, ya que desde EJsS se pueden realizar todas las labores anteriormente indicadas. Aún más, no sólo permite descargar y modificar el código del controlador en EJsS, si no que también permite que se descargue el makefile utilizado para compilarlo.

O1.C) *Posibilitar el volcado de la página web de HTML5.0 y javascript creada en EJsS por el responsable del laboratorio para que el estudiante pueda interactuar desde su PC, móvil o tableta con la experiencia que debe realizar.* Este objetivo también se ha logrado, ya que desde EJsS se puede descargar el entorno gráfico de la experiencia desarrollado en EJsS sobre la Raspberry PI, y hacer que este sea directamente reutilizable por el servidor en la próxima conexión del alumno.

O1.D) *Configurar y descargar la página web de instrucciones del laboratorio remoto.* Este objetivo también se ha logrado, ya que a la vez que se despliega la interfaz gráfica de la herramienta, se descarga la página de Descripción de la misma que ha sido desarrollada en EJsS.

O1.E) *Configurar la información que debe ser almacenada en los ficheros de datos de cada experimento.* Este objetivo también se ha conseguido, ya que en el fichero de configuración de las variables que se intercambian con el servidor, al que se tiene acceso desde EJsS también se indican cuales deben ser almacenadas en el fichero de datos de cada experiencia.

O1.F) *Simplificar el proceso de alta y baja de usuarios (alumnos) a la página web del laboratorio.* Este objetivo también se ha logrado, ya que desde EJsS se pueda dar de alta/baja a diferentes alumnos, bien de forma manual (de uno en uno) o a través de un archivo CSV.

Por lo tanto, se han cumplido todos los objetivos del primer grupo, y la nueva metodología permite centralizar la gestión del laboratorio desde EJsS. Aun más, conviene resaltar que no se ha hecho una modificación ad-hoc de la funcionalidad y el entorno gráfico de EJsS, sino que se ha ido un paso más allá: se ha dotado a EJsS con una nueva funcionalidad que permite a cualquier usuario con conocimientos de programación e interés en adaptar EJsS a sus necesidades, realizar sus modificaciones y encapsularlas en un "Plugin" propio, que puede ser distribuido fácilmente y

reutilizable en nuevas versiones de EJsS. Esta modificación de EJsS, inicialmente no contemplada en este proyecto, ha sido incorporada a partir de la versión oficial 6.0 de dicha herramienta, y está sirviendo para articular las nuevas versiones de dicha herramienta en conjuntos de funcionalidades diferentes para distintos tipos de usuarios.

O2. Extensión del uso de la herramienta a experiencias de diferentes asignaturas

O2.A) *Documentar el funcionamiento de la nueva herramienta, para que los responsables de los laboratorios dispongan de la información necesaria para crear desde cero un laboratorio remoto.* Se han desarrollado dos manuales, uno en el que se indica el proceso de instalación del software necesario para poder desarrollar los laboratorios remotos y otro en el que se presenta el proceso seguido para poner a punto un laboratorio desde cero. De forma complementaria, también se han desarrollado unos vídeos en los que se detallan las partes más relevantes del proceso. También se han documentado la mayoría de las experiencias que se han actualizado/desarrollado a lo largo del PIMCD, para que sirvan de ejemplo a posibles usuarios de la herramienta.

O2.B) *Pre-seleccionar y poner a punto los laboratorios remotos de un conjunto experiencias del área de Ingeniería de Sistemas y Automática (ISA) y de Física del Estado Sólido.* Este objetivo se ha conseguido de forma parcial. Se ha puesto apunto (actualizando o desarrollando desde su comienzo) 5 experiencias de ISA, relacionadas con las asignaturas de Sistemas Lineales, Procesamiento de Señales y Control de Sistemas, y se ha desarrollado el software necesario para que los alumnos también puedan interactuar con un brazo robótico a través de Internet desde sus equipos. Queda pendiente desarrollar las experiencias concretas que se realizarán sobre el brazo robótico y otras relacionadas con el área de Física del Estado Solido. Estas últimas han sido pospuestas, ya que la conexión de los dispositivos de las experiencias que se deseaba poner en remoto con la Raspberry PI se ha visto dificultada por la restricciones de la pandemia.

O2.C) *Publicitar y publicar la nueva herramienta que se desarrolle en los foros adecuados..* Para lograr este objetivo se han realizado dos publicaciones. La primera es una comunicación a las XL Jornadas de Automática, dentro de la sesión del grupo de Educación en Control del Comité Español en Automática (CEA). La segunda es un artículo publicado en el número especial "Open Source Software in Learning Environments" de la revista internacional *Electronics*. Además, en la actualidad se están preparando dos publicaciones adicionales: un nuevo artículo, que será próximamente enviado a la revista internacional *IEEE Transactions on Educations* (que tiene como objeto presentar de forma detallada la versión actual de la metodología que se ha actualizado en este PIMCD para gestionar nuestros laboratorio remotos de bajo coste desde EJsS.) y una nueva comunicación a las próximas Jornadas de Automática, relacionada con los desarrollos del laboratorio remoto sobre el brazo robótico.

Por otra parte, es importante destacar que se ha creado un repositorio desde el que se puede acceder al "Plugin" de los laboratorios remotos, al código del servidor, al código del software de control de los sistemas reales, y los proyectos de EJsS desde los que se desarrollan las interfaces gráficas de las experiencias y se gestionan de forma centralizada los laboratorios disponibles.

Cabe indicar que ha quedado pendiente la difusión de la existencia de la herramienta dentro del ámbito más próximo, el de la propia Facultad de Ciencias Físicas, para explorar la posibilidad de desarrollar nuevas experiencias con esta metodología en un futuro próximo.

Por lo tanto, podemos concluir que los objetivos de este grupo no han sido cumplidos plenamente, principalmente debido a las dificultades que la pandemia ha añadido a nuestras labores docentes y de investigación habituales. Aun así, podriamos considerar que en conjunto se ha logrado el espiritu de los mismos, ya que se está consiguiendo publicitar la herramienta a una amplia comunidad a través de las publicaciones y se ha extendido su uso a dos nuevas asignaturas (Procesamiento de Señales y Robótica) en la que no estaban siendo utilizada previamente.

3. Metodología empleada para el proyecto

La metodología empleada para el proyecto ha sido, a grandes rasgos, la siguiente:

- Se ha modificado la herramienta EJsS para incorporarle una nueva funcionalidad, de uso genérico, que permite a cualquier usuario con conocimientos avanzados de programación y del funcionamiento interno de la herramienta, modificar su funcionalidad y su entorno gráfico mediante un "Plugin". Esta nueva capacidad de EJsS, que ha sido integrada en la herramienta a partir de su versión 6.0, no es únicamente válida para este proyecto, si no que puede ser utilizada por otros usuarios de la herramienta. Aunque no era un objetivo inicial del Proyecto, constituye una de las aportaciones más relevantes del mismo.

- Se ha desarrollado el "Plugin" correspondiente para permitir que desde EJsS se pueda gestionar el laboratorio remoto de bajo coste basado en EJsS, Node.js y la Raspberry PI. De forma simultánea, se ha ido modificando la funcionalidad del servidor del laboratorio, ya que además de la funcionalidad inicial con el que estaba dotado (relacionado con el despliegue de la página web del laboratorio) se ha tenido que integrar toda la funcionalidad necesaria para que el servidor de respuesta a las peticiones que desde EJsS se realizan para configurar el funcionamiento de nuevas prácticas. Por lo tanto, en la actualidad se dispone de un nuevo servidor y de un Plugin para el laboratorio remoto de EJsS.

- Se han actualizado/desarrollado 5 nuevas experiencias para las asignaturas de Sistemas Lineales, Procesamiento de Señales y Control de Sistemas del grado de Ingeniería Electrónica en Comunicaciones. Estas experiencias parten, en diferente grado, de algunas experiencias previas que ya estaban siendo utilizadas por los alumnos

- Se ha desarrollado desde cero un nuevo laboratorio remoto con un brazo robótico. Este nuevo laboratorio remoto servirá de punto de partida para realizar nuevas experiencias remotas en las asignaturas de Robótica de la Facultad de Informática y de la Facultad de Ciencias Físicas.

- Se ha documentado de forma detallada los diferentes elementos que constituyen el laboratorio: la metodología de los Plugins, el nuevo servidor del laboratorio y su Plugin, las nuevas experiencias realizadas, el proceso de instalación del laboratorio, su puesta en marcha desde cero, y buena parte de las experiencias remotas actualizadas/desarrolladas en el proyecto. La documentación generada, se anexa a esta memoria resumen inicial, ya que en ella se detalla mucha información relevante que puede ser util para los nuevos usuarios de la metodología.

- Durante el proceso, se han escrito y publicado dos artículos: uno a las Jornadas de Automática y otro al número especial "Open Source Software in Learning Environments" de la revista internacional *Electronics*. Además, se están preparando dos artículos adicionales, que serán enviados próximamente.

- La metodología ha sido probada por los desarrolladores directos de la misma, por los profesores del proyecto y por algunos alumnos que la han utilizado durante el desarrollo sus Trabajos de Fin de Grado.

Por último, conviene indicar que el proceso anterior no se ha realizado de forma secuencial, si no que se han ido desarrollando en paralelo diferentes etapas, de forma que los desarrolladores de la herramienta han podido recibir feedback de los usuarios de la misma.

4. Recursos humanos

Los recursos humanos del proyecto, y las labores que han realizdo cada uno de ellos, han sido los siguientes:

Los profesores Eva Besada, José Antonio López y Jesús Chacón han sido los responsables de desarrollar y programar la nueva metodología de los laboratorios remotos, debido a la participación de los dos primeros en el desarrollo de la herramienta existente, a la destreza del tercero en la realización de extensiones (complementos/plugins/elementos) para EJsS, y a la experiencia de los tres en el desarrollo de diferentes metodologías para la creación de laboratorios remotos. Además serán responsables de desarrollar los diferentes manuales, de documentar el trabajo en sus diferentes aspectos, y de supervisar el trabajo de los alumnos que colaboren en la puesta en marcha de nuevas prácticas.

Los profesores Juan Francisco Jiménez y Segundo Esteban han sido los responsables de actualizar alguna de las prácticas del laboratorio remotos (debido a su experiencia previa en el uso y en la puesta en marcha de otros laboratorios remotos de ISA de la UCM) y de proporcionar una realimentación respecto al funcionamiento de la herramienta. Esta experiencia resulta fundamental para poder usar una herramienta antes de que se encuentre documentada.

Las profesoras Ana Urbieta y Paloma Fernández eran las responsables de las prácticas de Materiales (que no se han podido desarrollar debido a la COVID-19). y de proporcionar la retroalimentación critica necesaria para mejorar la herramienta y la información disponible. Por lo tanto, su labor se ha centrado en estos últimos aspectos: han ido revisando la documentación que se estaba generando y proporcionado una realimentación útil sobre aspectos que consideraban interesantes añadir en las diferentes fase de desarrollo.

El personal técnico compuesto por David Sanchez y Felix Marquez ha proporcionado el apoyo necesario para realizar diferentes labores, como la instalación de las herramientas necesarias, el montaje del hardware de las nuevas prácticas, y la conexión de los equipos a la red.

Los alumnos que inicialmente iban a participar en el desarrollo del proyecto, han realizado algunas labores de prueba de la herramienta a nivel de usuario. Sin embargo, no han podido desarrollar una práctica con la misma debido a la COVID-19. Sin embargo, durante la prorroga del proyecto y durante el curso 2020-21 se han incorporado dos nuevas alumnas, Daniela Alejandra Goncalves Lopez de Medrano y Amalia Pintado Barrera, que han desarrollado sus Trabajos Fin de Grado con la herramienta y colaborado en la puesta en marcha parte de las experiencias disponibles al final del proyecto

5. Desarrollo de las actividades

Para crear la nueva herramienta integral de desarrollo de laboratorios remotos y extender su uso a diferentes experiencias de Ciencias e Ingeniería se han realizado las siguientes tareas:

Tarea 1) Determinado los mecanismos más adecuados que EJsS incorpora para realizar las diferentes extensiones requeridas por nuestra herramienta integral. Estas extensiones serán, con el objeto de dar respuesta al Objetivo 1 del proyecto, las siguientes: extensión de conexión entre variables (O1.A); extensión de la edición, volcado y compilación del código C (O1.B); extensión del volcado de la página web de visualización de la experiencia y de la de descripción de la misma (O1.C y O1.D), extensión para determinar las variables almacenadas en el fichero de datos (O1.E), y extensión para configurar el acceso de los alumnos al laboratorio (O1.F).

Mas concretamente, como non era trivial introducir de la forma que considerabamos más adecuada todos las funcionalidades y cambios de interfaz en EJsS asociadas a cada uno de los objetivos de este proyecto, durante el proyecto nos planteamos dotar a EJsS con la funcionalidad necesaria para que sus usuarios con conocimientos avanzados de programación puedan adaptar la herramienta a sus necesidades concretas. Este planteamiento es el origen que la nueva metodología de Plugins incorporada en EJsS, y publicada en el número especial "Open Source Software in Learning Environments".

Tarea 2) Implementado los complementos (plugins, elementos) de JAVA necesarios para que EJsS pueda incorporar cada uno de los mecanismos de extensión anteriormente mencionados.

Más concretamente, estos complementos de EJsS se han implementado y encapsulado en un nuevo Plugin de EJsS. Este plugin permite que los usuarios de EJsS puedan disponer de una interfaz y funcionalidad adecuada para llevar a cabo desde EJsS todas las tareas necesarias para poner en marcha y/o realizar el mantenimiento de nuestros laboratorios remotos.

Tarea 3) Modificado el servidor de Node.js para que pueda dar soporte a las peticiones realizadas desde EJsS para cada uno de sus mecanismos de extensión.

De hecho, debido a esta tarea, se dispone de un servidor de prácticas con dos tipos de funcionalidades diferente. La primera, y más relevante, es servir todas las páginas web del laboratorio (i.e. la página de acceso de usuarios, la de descripción de la práctica, la de interfaz de la experiencia, y la de acceso a los datos de las experiencias de cada alumno). Esta funcionalidad, que existía en la versión previa del laboratorio, ha sido mejorado y organizada para facilitar la gestión de los laboratorios. La segunda es la que permite que todas las peticiones que los docentes realizan desde EJsS para configurar las experiencias, sean atendidas por el servidor, y por lo tanto, son el soporte necesario para poder llevar a cabo dichas labores con éxito. Es decir, para poder aprovechar realmente la funcionalidad del Plugin desarrollado para EJsS ha sido necesario que el servidor del laboratorio incorpore la funcionalidad correspondiente para llevar a cabo cada una de las nuevas labores integradas en el Plugin.

Tarea 4) Actualizado las prácticas remotas existentes de bajo coste del laboratorio de ISA que utilizan la versión actual de la metodología EJsS+Node.js+raspberry PI.

Más en concreto, se han actualizado las práctica de caracterización de circuitos analógicos que estaban siendo utilizadas en las asignaturas de Sistemas Lineales de Ingeniería Electrónica de Comunicaciones.

Tarea 5) Desarrollado nuevas experiencias remotas del área de ISA (de Robótica y de Procesamiento de Señales).

Las prácticas adicionales de han sido elegidas entre las asignaturas del área en las que imparten docencia los profesores involucrados en este proyecto. En concreto, son prácticas/experiencias de Procesamiento de Señales, Control de Sistemas y Robótica.

Tarea 6) Documentado a) la estructura y funcionamiento general de nuestros laboratorio remotos, b) el uso de la herramienta integral de definición y configuración de los mismos, y c) todas las experiencias desarrolladas.

Para ello, se ha generado diferente documentación, en forma de información genérica de este proyecto, manuales para los profesores y guiones de prácticas para los alumnos. Toda esta documentación, se anexa, como una versión extendida de este documento, a continuación del mismo.

Tarea 7) Publicado información referente a la herramienta desarrollada en el mundo académico educativo.

Más concretamente, se ha realizado una primera publicación en las Jornadas de Automática del 2019, a las que asiste de forma habitual un grupo de investigadores centrados en el desarrollo de herramientas/laboratorios educativas para laboratorios virtuales y remotos, y en el número especial "Open Source Software in Learning Environments" de la revista internacional *Electronics* (ya que EJsS es una herramienta de código abierto extensamente utilizada en el área de Ciencias). Además, se están preparando dos publicaciones adicionales con ls resultados de este proyecto que serán enviados próximamente a otra revista internacional y a las Jornadas de Automática del 2021.

6. Anexos

A continuación se anexa la documentación detallada de todo el proyecto, como un documento autocontenido en el que se describen los objetivos, la metodología seguida para incorporar el mecanismo de Plugins en EJsS y para desarrollar la nueva versión del laboratorio, todas las experiencias actualizadas/desarrolladas, diferentes manuales para los profesores del laboratorio, y los guiones de la mayor parte de las experiencias realizadas.

UNIVERSIDAD COMPLUTENSE DE MADRID

PROYECTO DE INNOVACIÓN
Convocatoria 2019/2020

Número 139

Herramienta integral y de bajo coste para el desarrollo de prácticas remotas para las asignaturas de Ciencias e Ingeniería

Coordinadora: Eva Besada Portas

Facultad de Ciencias Físicas

Departamento de Arquitectura de Computadores y Automática

Herramienta integral y de bajo coste para el desarrollo de prácticas remotas para las asignaturas de Ciencias e Ingeniería

Resumen:

Los laboratorios remotos, en los que los alumnos interactúan con los dispositivos reales a través de Internet, incrementa el tiempo de uso de dichos dispositivos por parte de los alumnos y el número de pruebas que pueden realizar durante las experiencias. Sin embargo, para implementarlos, los docentes habitualmente tienen que saber utilizar una serie de tecnologías diferentes, que dificultan su puesta en marcha y mantenimiento.

Para paliar estos problemas, en este Proyecto de Innovación y Mejora de la Calidad Docente (PIMCD) se ha desarrollado una nueva metodología desde la que se centraliza la puesta en marcha de todo el laboratorio desde una única herramienta (Easy JavaScript Simulations, EJsS), que sirve tanto para gestionar el código mediante el que se interactúa con los dispositivos, el servidor que da acceso remoto a la práctica, y la interfaz gráfica con la que los alumnos interactúan con los dispositivos. Además, es importante resaltar que es una metodología de bajo coste, ya que desde el punto de vista del proceso de conversión de un laboratorio presencial ya existente en uno remoto, solo necesita una Raspberry PI para desplegar todo el software del laboratorio.

Además, en este proyecto también se presentan un conjunto de experiencias desarrolladas con las mismas, que ya están siendo utilizadas por los alumnos en diferentes asignaturas de Ingeniería Electrónica de Comunicaciones y un nuevo laboratorio remoto de Robótica que se pondrá en marcha el curso que viene.

Índice general

Capítulo 1

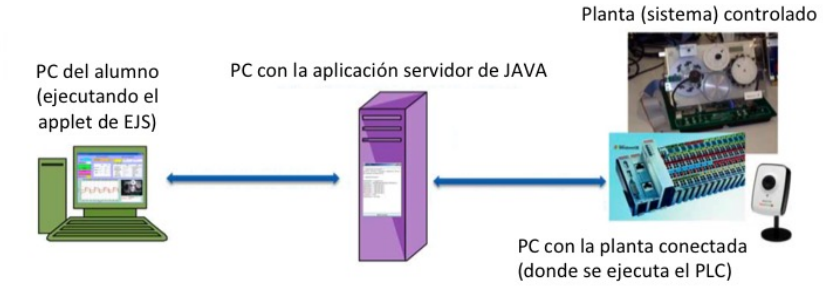
Introducción

Las asignaturas de Ciencias e Ingeniería tienen una carga experimental importante, que se pone de manifiesto en las sesiones de laboratorio, donde los alumnos interactúan con diferentes dispositivos, que les permiten aplicar sus conocimientos teóricos sobre diferentes experimentos reales.

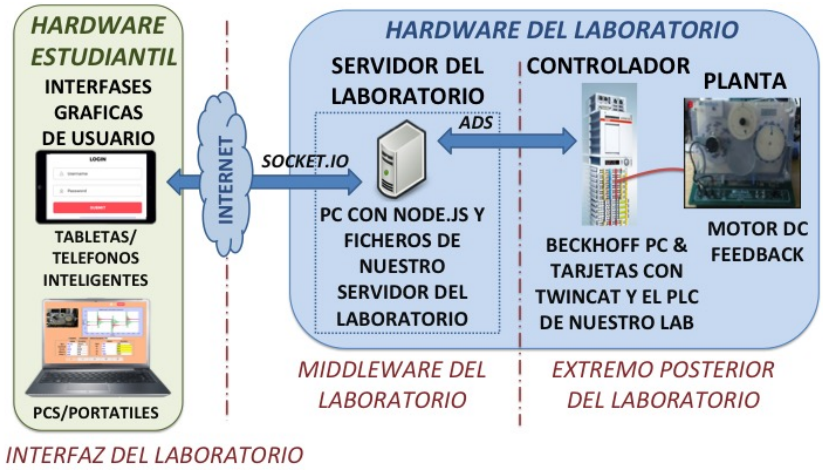
La realización de este tipo de experiencias es fundamental, ya que no sólo permite a los alumnos observar los fenómenos que han estudiado y aprender los métodos de trabajo de su titulación, sino que también les ayuda a asimilar sus principales aspectos y conclusiones teóricas. Sin embargo, el coste del material de laboratorio reduce habitualmente el número de puestos experimentales disponibles, limita el tiempo de experimentación de los alumnos durante las sesiones presenciales de laboratorio y restringe sus posibilidades de aprendizaje. Los laboratorios remotos, que dan acceso a través de internet, a las experiencias de los laboratorios tradicionales, eliminan los inconvenientes anteriormente mencionados y expanden las posibilidades de aprendizaje del alumno (tanto antes como después de las sesiones presenciales del laboratorio).

Por todos estos motivos, el grupo de Investigación de Ingeniería de Sistemas, Control, Automatización y Robótica (ISCAR) de la Universidad Complutense de Madrid (UCM) participa de forma activa en el grupo de "Educación" del Comité Español de Automática (CEA), desarrollando, desde al año 2010, diferentes metodologías para construir laboratorios remotos para las asignaturas de Control del área de Ingeniería de Sistemas y Automática (ISA) de la UCM. La diferencia entre las metodologías desarrolladas se resume en las gráficas de la Figura 1.1, en las que se resaltan los equipos necesarios en cada caso y las herramientas utilizadas por cada una de las metodologías que se describen brevemente a continuación.

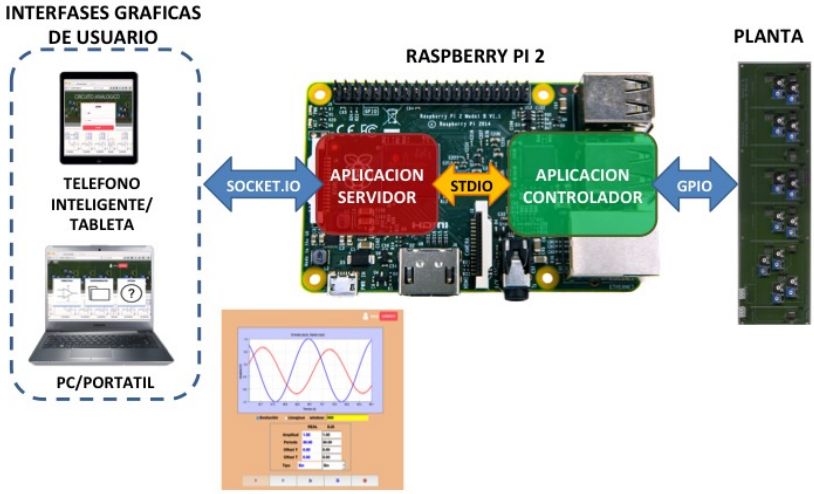
La primera de las metodologías desarrolladas, con la financiación del Proyecto de Innovación Educativa PIMCD-UCM-2011-211 y del PIMCD-UCM-2014-318 [1], ha sido publicada en [2] y se basa, tal y como se esquematiza en la Figura 1.1a, en el uso de Controladores Lógicos Programables (PLCs) para la interacción directa con los dispositivos reales, de applets desarrollados con la herramienta de programación gráfica Easy Java Simulations (EJS, [3]) para que los alumnos accedan a la experiencia a través de internet, y de un servidor de JAVA para intercambiar información entre los PLCs y los applets. Su uso continuado en diferentes prácticas de la UCM y dentro del portal UNILabs, finalizó en 2016, como consecuencia a las restricciones impuestas por los navegadores de internet al uso de applets desde una página web. Para resolver este problema, el grupo ISCAR ha desarrollado dos nuevas metodologías desde entonces. En una de ellas, esquematizada en la Figura 1.1b y publicada en [4], se sustituyen el applet de EJS y el servidor JAVA, por una pagina web HTML5 con javascript generada con la herramienta Easy Java JavaScript Simulations (EJsS, sucesora de EJS) y un servidor de Node.js, y se mantienen los PLCs previos. Por lo tanto, puede considerarse la versión actualizada de la primera metodología, con la capacidad añadida de dar un servicio integral a todas las páginas web del laboratorio remoto correspondiente. En la otra, esquematizada en la Figura 1.1c y publicada en [5, 6], se utiliza también el fichero HTML5 con



(a) Interfaz Applet EJS+ servidor JAVA + Controlador PLC



(b) Interfaz Pagina Web + servidor Node.js + Controlador PLC



(c) Interfaz Pagina Web + servidor Node.js + controlador en C

Figura 1.1: Esquemas de las metodologías previamente desarrolladas por el grupo ISCAR

javascript definido mediante EJsS y el servidor de Node.js, y se sustituye el PLC por un computador de bajo coste (una Raspberry PI) con funciones de entrada-salida en el que se programa en C las funciones que interactúan con los dispositivos reales y en el que se despliega el servidor de Node.js. Por lo tanto, constituye una solución compacta de coste mínimo (respecto a la conexión a internet y al acceso remoto del laboratorio), que puede ser utilizada en numerosas experiencias de las asignaturas de las áreas de Ciencias e Ingeniería.

Sin embargo, aunque la última de las tecnologías mencionadas permite dar acceso remoto a los equipos del laboratorio con un coste reducido, requiere que los profesores y/o técnicos responsables de poner en marcha las experiencias, utilicen múltiples herramientas (EJsS, Node.js, editor y compilador del código en C) para poder hacerlo. Por lo tanto, dificulta la extensión de su uso a aquellas asignaturas en las que los responsables no están familiarizado con todas estas herramientas.

Por este motivo, este proyecto tiene como objetivos: 1) automatizar el proceso de generación de los laboratorios remotos de bajo coste de la UCM basados en EJsS, Node.js y la Raspberry PI; para 2) intentar extender su uso a diferentes asignaturas del área de Ciencias e Ingeniería.

Para alcanzarlos, por una parte, se incluirán nuevas funcionalidades en EJsS, de forma que esta herramienta (inicialmente solo responsable de generar la página web que utilizan los alumnos como interfaz directo de los dispositivos del laboratorio) sirva para llevar a cabo una gestión integral del laboratorio, incluyendo entre sus nuevas labores: el despliegue automático de la página web del laboratorio en la Raspberry PI, la edición y compilación del programa controlador (responsable de la interacción directa con los dispositivos reales), y la gestión de usuarios del laboratorio. Por otra, se intentará extender su uso a diferentes tipos de experiencias. Es importante indicar que este proceso de extensión, a su vez, servirá para que la nueva herramienta que se desarrolle durante este Proyecto de Innovación, sea puesta a prueba por personal ajeno a su desarrollo, y por lo tanto, el proceso se beneficie de la realimentación obtenida por los nuevos usuarios.

Finalmente hay que tener en cuenta que existen otras metodologías para el desarrollo de laboratorio remoto que aúnan las herramientas que han sido utilizadas a lo largo de este proyecto. Por ejemplo, la metodología descrita en [7, 8] utiliza: 1) una Raspberry PI con un servidor en Node.js y un programa en un lenguaje de programación estándar para dar acceso a través de la pagina web desarrollada con EJsS a la experiencia con los dispositivos reales y 2) un servidor Moodle adicional (alojado en otro equipo) para proporcionar el resto de los servicios web del laboratorio remoto (p.e. control de acceso y simulaciones). Por lo tanto, la herramienta de este proyecto será una solución más económica y compacta, al alojar al servidor completo (en Node.js) de páginas web del laboratorio remoto, la interfaz desarrollada en EJsS y el código del controlador en una Raspberry PI.

1.1. Objetivos

El objetivo principal de este proyecto es el desarrollo de una herramienta integral y de bajo coste para dar acceso remoto, a través de Internet, a las experiencias de los laboratorios de las áreas de Ciencias e Ingeniería. Como herramienta integral entendemos que ésta debe permitir realizar todas las labores necesarias para poner en marcha las experiencias del laboratorio desde un mismo entorno. Con su bajo coste indicamos que el gasto asociado a conectar un laboratorio existente a internet no sea, salvo por necesidades de sensorización y actuación especiales, superior a los 150 €.

El objetivo general se puede articular en los siguientes dos objetivos diferenciados y en sus correspondientes subconjuntos de propósitos concretos:

Objetivo 1 (O1). Automatizar e integrar en una única herramienta el proceso de generación de los laboratorios remotos de bajo coste de la UCM basados en EJsS, Node.js y la Raspberry

PI. Para alcanzarlo, se extenderá las funcionalidad de la herramienta de programación gráfica EJsS con las siguientes capacidades:

O1.A) Facilitar la conexión entre las variables utilizadas en el código del controlador (que interactúa con los sensores y actuadores conectados al dispositivos) y las variables javascript de la página web que da acceso al laboratorio. Es decir, la herramienta EJsS extendida dispondrá de la funcionalidad necesaria para que el usuario pueda conectar de forma automática las variables que el código C envía al servidor con las variables que la página web del laboratorio utiliza para que los alumnos puedan visualizar la evolución del estado del sistema real e interactuar con él.

O1.B) Permitir acceder y editar el código del controlador de la Raspberry PI desde EJsS, y una vez que esté listo, descargarlo y compilarlo desde EJsS sobre la Raspberry PI. Es decir, la herramienta extendida EJsS dispondrá de la funcionalidad necesaria para evitar que el usuario tenga que utilizar herramientas adicionales para modificar y volcar el código más cercano a la experiencia, y lo almacenará como parte del proyecto del laboratorio remoto que ha creado, para evitar la disparidad de versiones entre lo que está volcado en la Raspberry PI y lo que está almacenado en el proyecto de EJsS.

O1.C) Posibilitar el volcado de la página web de HTML5.0 y javascript creada en EJsS por el responsable del laboratorio para que el estudiante pueda interactuar desde su PC, móvil o tableta con la experiencia que debe realizar. Es decir, se eliminará una etapa farragosa del proceso actual, en la que el responsable debe: 1) extraer, del código generado desde EJsS, la parte necesaria para generar la página web de la experiencia; 2) modificarlo ligeramente; y 3) descargarlo sobre los directorios pertinentes de la Raspberry PI.

O1.D) Configurar y descargar la página web de instrucciones del laboratorio remoto. Es decir, el responsable del laboratorio configurará la información necesaria para cada experiencia desde EJsS y ésta será automáticamente descargada sobre la Raspberry PI para que pueda ser utilizada como página web de ayuda dentro del laboratorio remoto.

O1.E) Configurar la información que debe ser almacenada en los ficheros de datos de cada experimento. Es decir, el responsable del laboratorio dejará de tener que modificar el código del servidor Node.js para que éste grabe la información apropiada de cada experimento, utilizando alternativamente para éste fin una nueva funcionalidad que incorporará la versión extendida de EJsS para facilitar y agilizar la puesta en marcha de los laboratorios remotos de bajo coste de la UCM.

O1.F) Simplificar el proceso de alta y baja de usuarios (alumnos) a la página web del laboratorio, por medio de una nueva funcionalidad en EJsS que permita al responsable del laboratorio dar fácilmente de alta y baja a aquellos alumnos que pueden acceder a la experiencia en diferentes instantes. De esta forma, se agiliza otra etapa del proceso actual, en la que el responsable debe modificar de forma manual una base de datos alojada en la Raspberry PI.

Objetivo 2 (O2). Extender el uso de la herramienta a las experiencias de laboratorio de diferentes asignaturas de Ciencias e Ingeniería. Para lograrlo, se deberán realizar las siguientes labores:

O2.A) Documentar el funcionamiento de la nueva herramienta, para que los responsables del laboratorio dispongan de la información necesaria para crear desde cero un laboratorio remoto. Este propósito resulta fundamental para extender el uso de la herramienta más allá de los profesores involucrados en su desarrollo.

O2.B) Pre-seleccionar y poner a punto los laboratorios remotos de un conjunto experiencias de diferentes campos, teniendo en cuenta las áreas de conocimiento en la que imparten docencia los participantes del proyecto. Más en concreto, se piensa actualizar las prácticas de Control de bajo coste ya disponibles a la nueva metodología, y desarrollar nuevas experiencias de Robótica y Procesamiento de Señales (ambas afines al área de ISA) y de Física del Estado Sólido (afín al área de Materiales).

O2.C) Publicitar y publicar la nueva herramienta que se desarrolle en los foros adecuados. Por una parte, una vez que estén listos los diferentes laboratorios se planea difundir su existencia dentro del ámbito más próximo, el de la propia Facultad de Ciencias Físicas, para explorar la posibilidad de desarrollar nuevas experiencias con esta metodología en un futuro próximo. Por otra, se desea publicar los resultados obtenidos en jornadas, conferencias y revistas relacionadas con el desarrollo de nuevas herramientas educativas para la creación de laboratorios remotos (como podrían ser las Jornadas de Innovación de la UCM, las Jornadas de Automática, expAt o el IEEE Transaction on Education).

En conjunto, los objetivos propuestos se centran en torno a la necesidad de disponer de herramientas integrales que automaticen y agilicen el proceso de puesta a punto de laboratorios remotos a partir de experiencias existentes en laboratorios presenciales, y de extender el uso de este tipo de laboratorios dentro de asignaturas que se puedan beneficiar de su existencia en los campos de Ciencias e Ingeniería, y en nuestro entorno más cercano.

Finalmente, conviene indicar que la herramienta de bajo coste para la creación de laboratorios remotos que se desarrollará en este proyecto resulta especialmente útil en la situación epidemiológica en la que nos encontramos por la COVID-2019, ya que puede permitir que los alumnos realicen desde sus casas, sin tener que desplazarse a sus centros de estudios, parte de las experiencias de los laboratorio presenciales tradicionales.

1.2. Organización de la memoria

A continuación describimos la organización del resto de la memoria de este PIMCD.

En el Capítulo 2 se describe la metodología seguida para poder centralizar la gestión del laboratorio remoto a través de la herramienta EJsS. Más concretamente, el capítulo comienza presentando la metodología desarrollada para poder incorporar nuevas funcionalidades y elementos al interfaz gráfico de EJsS, y continúa detallando las mejoras concretas introducidas en EJsS para gestionar el laboratorio remoto.

En el Capítulo 3 se presentan todas las experiencias que se han adaptado o desarrollado con la nueva metodología, organizadas de acuerdo con el dispositivo sobre el que se desarrollan. Más concretamente, primero describe las experiencias desarrolladas sobre un circuito analógico reconfigurable para las asignaturas de Sistemas Lineales, Procesamiento de Señales, y Control de Sistemas de la titulación de Ingeniería Electrónica en Comunicaciones, y a continuación presenta el laboratorio remoto que se ha desarrollado sobre un brazo robótico para diferentes asignaturas de la Facultad de Informática y Ciencias Físicas.

En el Capítulo 4 se presentan las conclusiones de este trabajo, revisando el grado de cumplimiento de los diferentes objetivos y presentando alguna lineas de trabajo futuras.

Tras el cuerpo principal del documento, se presenta un conjunto de apéndices en el que se documentan: el proceso de instalación del software necesario para desarrollar y desplegar nuevos laboratorios remotos, el procedimiento a seguir para desarrollar un nuevo laboratorio remoto desde EJsS, los pasos que los alumnos han de seguir para acceder al laboratorio remoto, el circuito reconfigurable y de adecuación de señales que se utiliza en diferentes experiencias, las experiencias de

la asignatura de Sistemas Lineales (identificación del modelo de un sistema mediante su respuesta temporal y mediante su diagrama de Bode experimental), la experiencia de la asignatura de Procesamiento de Señales (identificación del modelo de un sistema mediante el estudio en frecuencia de su respuesta a una señal aleatoria), y las experiencias de la asignatura de Control de Sistemas (diseño de una red de adelanto y de reguladores PID -Proporcionales, Integrales y Derivativos).

Capítulo 2

Extendiendo la funcionalidad de EJsS para centralizar el desarrollo de laboratorios remotos

El primero objetivo del proyecto consiste en extender las capacidades de EJsS para convertirlo en la herramienta central, desde la que se realizarán todas las labores de creación, gestión y mantenimiento de nuestro laboratorio de bajo coste.

Para modificar la funcionalidad de esta herramienta gráfica de código abierto es necesario, como cabría esperar, llegar a conocer como se encuentra estructurada/programada y como se puede incorporar nueva funcionalidad en diferentes elementos de la herramienta. Como este proceso de aprendizaje es bastante costoso y puede ser necesario llevarlo a cabo en diferentes etapas del desarrollo de nuestro laboratorio, en vez de realizar las modificaciones directamente sobre la versión más actual de la herramienta, se decidió introducir en EJsS una funcionalidad más genérica, que denominaremos "Plugins" a partir de este momento y que da soporte a una modificación estandarizada y poco intrusiva de EJsS. Brevemente, la idea consiste en reemplazar una modificación directa de EJsS para nuestro caso particular (es decir, para dar un soporte centralizado a la creación y mantenimiento de nuestros laboratorios remoto de bajo coste) por una modificación multi-etapa: en la primera fase modificamos EJsS de forma genérica para que sea capaz de utilizar la funcionalidad que le proporcione una pieza de código externa denominada "Plugin" y en la segunda etapa desarrollamos el "Plugin" que incluye la funcionalidad específica requerida por nuestro laboratorio. En las dos secciones siguientes presentaremos los aspectos más relevantes de software desarrollado en cada etapa.

Finalmente, la última sección del capítulo también resume la documentación que se ha desarrollado, respecto a esta extensión de la funcionalidad de EJsS, para los usuarios de las nuevas herramientas y metodología.

2.1. Extendiendo la funcionalidad de EJsS mediante "Plugins"

En esta sección se describen las principales características de EJsS (i.e. de la herramienta elegida como nexo de unión entre todas las herramientas software del laboratorio), se justifican los beneficios de la extensión mediante "Plugins", y se introducen los tipos de extensión soportados actualmente por las modificaciones realizadas.

Detalles adicionales sobre la extensión de EJsS mediante "Plugins" pueden encontrarse en [9], que es una publicación resultante de la metodología y el software desarrollado en este proyecto.

2.1.1. Descripción de EJsS

EJsS [3] es una herramienta de código abierta, que permite desacoplar la definición del *Modelo* de simulación y la *Vista* que muestra gráficamente la evolución de una simulación del modelo e interactuar con él. A partir del Modelo, habitualmente definido por el usuario con ecuaciones diferenciales ordinarias, y la Vista, especificada a partir de los diferentes elementos (gráficas, botones, etc.) que la constituyen, EJsS genera la página web, codificada en HTML + Javascript, que permite al usuario final simular el modelo e interactuar con él. Para simular el modelo, EJsS utiliza diferentes integradores numéricos (p.e. el método Euler o el algoritmo Cash-Karp), e integra procedimientos sofisticados de detección y manejo de eventos. Además, incorpora una paleta de elementos visuales e interactivos que permiten que sus usuarios diseñen entornos gráficos ilustrativos y atractivos para sus simulaciones. Por todos estos motivos, EjsS es una herramienta de gran utilidad para crear simulaciones interactivas en el área de Ciencias, para diseñar laboratorios virtuales y para desarrollar la *Vista* mediante la que los alumnos interactúan con los laboratorios remotos.

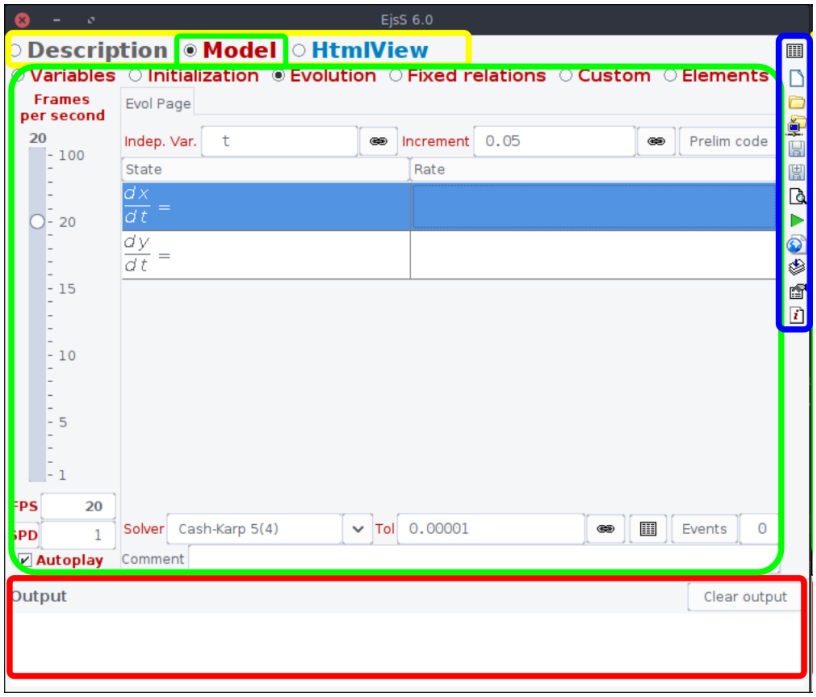
Para que los usuarios de EJsS puedan desarrollar fácilmente simulaciones o aplicaciones con esta herramienta, su funcionalidad se encuentra distribuida en varias áreas que agrupan elementos conceptualmente relacionados:

- El *Panel de Edición Principal*, situado en la parte superior de EJsS y resaltado en amarillo en las Figuras 2.1a y 2.1b, permite el acceso a las tres herramientas principales con las que se definen las simulaciones en EJsS. Por una parte, permite acceder al *Editor de Modelos*, que se encuentra resaltado en verde en la Figure 2.1a, mediante el que se definen las variables y ecuaciones diferenciales del modelo, se escriben las relaciones fijas y el código propio de cada modelo, y se pueden seleccionar Elementos (herramientas especiales) de EJsS. Por otra parte, el *Editor de la Vista*, resaltado en verde en la Figura Figure 2.1b, contiene, a su izquierda, el árbol de elementos visuales/interactivos mediante los cuales el usuario está definiendo la *Vista* de la simulación y, a su derecha, la paleta de elementos disponibles (e.g. botones, campos de texto, gráficas bidimensionales y elementos gráficos tridimensionales). Finalmente, el *Editor de la Descripción* se usa para que el usuario de EJsS escriba la página/información de ayuda que desea proporcionales a los usuarios finales de la simulación/aplicación desarrollada en EJsS.

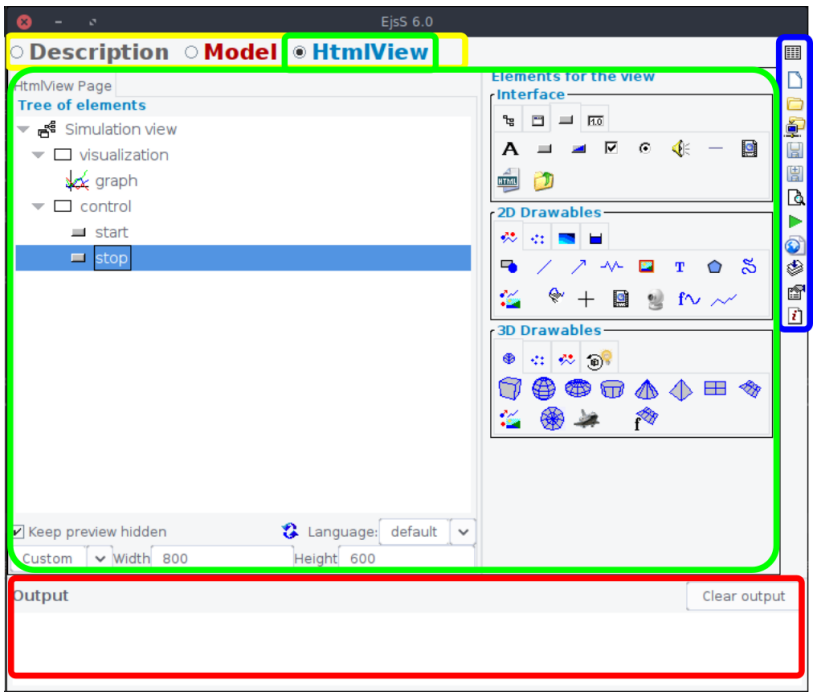
- La *Barra de Botones*, situada a la derecha de EjsS y resaltada en azul en las Figuras 2.1a y 2.1b, facilita el acceso a diferentes utilidades de uso habitual en EJsS (p.e. a la compilación o distribución de la aplicación, o la gestión del fichero del modelo que está siendo desarrollado). Además, cabe indicar que estos botones pueden hacer que se ejecute una acción determinada o que que se abra una ventada de dialogo que agrupa múltiples acciones.

- El *Área de Mensajes de Salida*, situada en la parte inferior de EJsS y resaltada en rojo en las Figuras 2.1a y 2.1b, es utilizada por EJsS para informar a sus usuarios sobre diferentes tipos de errores o situaciones que se están produciendo durante el uso de la herramienta.

Finalmente, cabe indicar que aunque la herramienta ha sido especialmente diseñada para el desarrollo de simulaciones interactivas para las asignaturas de Ciencia e Ingeniería, es utilizada para otros campos de aplicación, como el desarrollo de la *Vista* de numerosos laboratorios remotos o para realizar analíticas de aprendizaje. En el primer caso, se sustituye el modelo de simulación por las rutinas de comunicación entre la *Vista* y el servidor de las prácticas del laboratorio. Para el segundo, se ha incorporado una librería con herramientas estadísticas en EJsS. Ambos casos, en resumen, incorporan una nueva funcionalidad en EJsS, bien, de forma puntual en el código específico de cada aplicación EJsS (es decir, a través de la pestaña de código *Custom*izado) o de una forma más genérica a través de los Elementos de EJsS, que constituyen librerías de funcionalidad



(a) Pantalla mostrada tras elegir el Editor del Modelo



(b) Pantalla mostrada tras elegir el Editor de la Vista

Figura 2.1: Pantallazos del entorno gráfico estándar de EJsS

adicional únicamente utilizable en la definición del *Modelo* o del comportamiento de la aplicación. Sin embargo, la incorporación de nueva funcionalidad para la *Vista*, de nuevos *Editores* o *Botones* de acción requería, hasta la versión 5.3, una modificación directa del código de EJsS.

2.1.2. Justificación de la extensión de EJsS mediante "Plugins"

La modificación de la funcionalidad de EJsS siempre ha sido posible, ya que:

1. Es una **herramienta de código abierto** y, por lo tanto, modificable por cualquier usuario que así lo desee. Sin embargo, para poder realizar estas modificaciones es necesario tener un conocimiento bastante profundo de la estructura y funcionalidad del código de la herramienta, para que las modificaciones que se realicen no interfieran con la funcionalidad existente y logren el objetivo para el que han sido incorporadas. Además, las modificaciones incluidas de esta forma son difíciles de mantener/propagar a nuevas versiones de EJsS, ya que los cambios en la herramienta pueden tener sus implicaciones en el código incorporado por el usuario en la versión anterior.

2. Permite **incorporar nuevos Elementos Visuales** para satisfacer las necesidades sistemáticas de algún grupo de usuarios. Para lograrlo, el usuario de EJsS tiene que modificar localmente su versión de EJsS, implementar la interfaz *ViewElement* del nuevo elemento visual, asignarle un icono, y añadir el nuevo elemento en uno de las paletas de herramientas visuales existentes. Aunque está aproximación es menos intrusiva que la primera, el nuevo elemento visual solo está disponible en la versión de la herramienta de dicho usuario y deberá ser refactorizada para su inclusión en versiones futuras de EJsS.

3. Permite **incorporar nuevos Elementos de Modelo** que implementan comportamientos (e.g. las rutinas de comunicación de los laboratorios remoto o las herramientas estadísticas de análisis del aprendizaje) que pueden ser útiles para diferentes usuarios. Para este tipo de modificación EJsS proporciona una facilidad de extensión diferente a la del caso anterior: aunque es necesario implementar la interfaz *ModelElement*, el usuario en vez de incorporar el código directamente en su versión de EJsS, puede crear un fichero .JAR que incorpore la funcionalidad correspondiente, que estará disponible en la pestaña de Elementos del Editor del Modelo siempre y cuando el fichero .JAR se sitúe en la carpeta de elementos de EJsS. Esta forma de proceder es la que más se acerca a nuestro objetivo, ya que permite compartir fácilmente la nueva funcionalidad creada entre diferentes usuarios y reutilizarla, sin tener que modificar el código de EJsS, en las nuevas versiones de la herramienta.

Los mecanismos de modificación previos están organizados según un nivel decreciente de dificultad y un nivel creciente de reutilizabilidad del código en versiones futuras de la herramienta. Además, cabe preguntarse el motivo por el que los Elementos Visuales reciben un tratamiento diferente (a la hora de incorporarse en EJsS) al de los Elementos del Modelo. Aun más, como docentes que habitualmente usan EJsS para crear la interfaz gráfica de un laboratorio remoto echamos de menos ciertas funcionalidades que agilicen el proceso de puesta a punto de nuestros laboratorios y que podrían ser centralizadas desde EJsS, pero que realmente no son clasificables como Elementos de Modelo. De hecho nos preguntamos si sería posible añadir en la Barra de Botones una herramienta adicional que permita directamente descargar la interfaz gráfica creada en el servidor del laboratorio remoto. O si podría incorporarse un editor del código responsable de cerrar el lazo de control del laboratorio. O disponer de unos elementos visuales que reproducimos de forma sistemática en todas nuestras interfaces gráficas.

Para dar una respuesta genérica a estar preguntas, en vez de modificar directamente el código de EJsS para incorporar esas funcionalidad decidimos modificar EJsS para que sea capaz de utilizar la

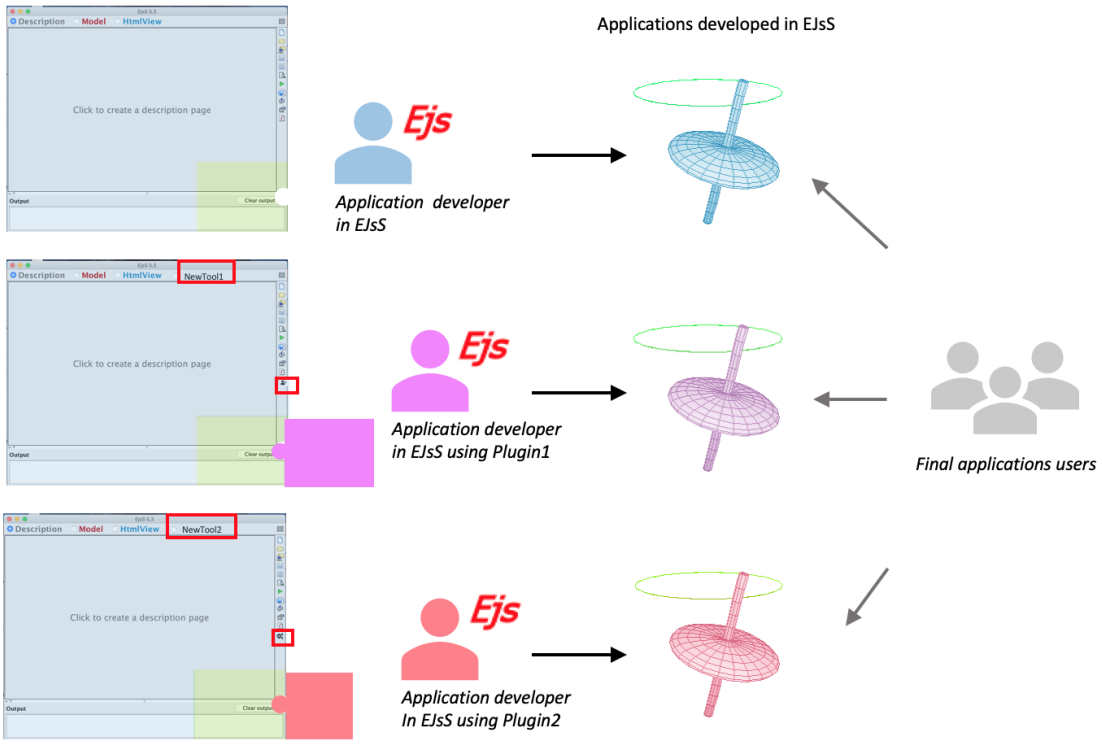


Figura 2.2: Modificación de EJsS mediante Plugins

funcionalidad que cualquier usuario considere adecuada para su aplicación particular, y generalizar el comportamiento de carga de Elementos del Modelo a otro tipo de Elementos (Editores, Elementos Gráficos, Botones) de EJsS. Esta nueva forma de incorporar funcionalidad en EJsS ha sido bautizada con el nombre de "Plugin", ya que tal y como se muestra en la figura esta página, la idea es que la funcionalidad y el entorno gráfico de EJsS se adapte a los diferentes ficheros JAR que el usuario conecte a la herramienta.

2.1.3. Objetivos y requisitos de los "Plugins"

El mecanismo de "Plugins" desarrollado a lo largo de este PIMCD tiene como objetivos:

- Proporcionar una estrategia elegante para facilitar la mejora de EJsS.
- Permitir que los desarrolladores con capacidades de programación avanzadas personalicen o amplíen la interfaz y la funcionalidad de EJsS.
- Crear un método de distribución de nuevas funcionalidades fácil de usar e independiente de las actualizaciones de EJsS.
- Permitir que los usuarios elijan qué extensiones de EJsS desean utilizar en cada momento.

Por otra parte, consideramos durante el desarrollo del nuevo mecanismo de "Plugins" que estos debían cumplir los siguientes requisitos:

- Distribución simple, en la que se permita descargar e instalar los "Plugins" fácilmente.

- Auto-contenido, de forma que el "Plugin" contenga todo el software necesario para que pueda ser cargado de forma automática por EJsS.
- No intrusivo, para poder revertir su instalación cuando se considere necesario.

2.1.4. Tipos de extensiones soportadas

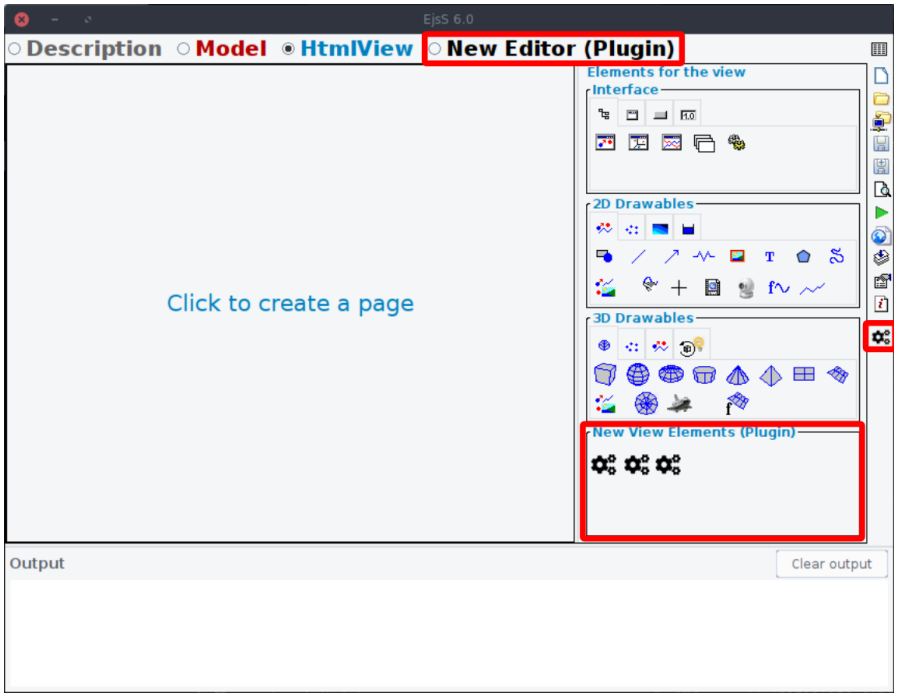
Después de un análisis detallado sobre qué funcionalidades y cambios en el interfaz queríamos integrar en EJsS, y varias entrevistas con otros usuarios habituales de la herramienta y con su desarrollador (el catedrático Francisco Esquembre), decidimos soportar las extensiones que se muestran en la Figura 2.3 y que se enumeran a continuación:

- *Nuevos Editores en el panel de Edición Principal*, tal y como se muestra en la parte superior izquierda de las Figuras 2.3a y 2.3b. El objetivo de este tipo de extensión es poder añadir otras funcionalidades genéricas, de alto nivel, que necesite un conjunto de usuarios concretos de EJsS.
- *Nuevos Elementos Visuales* disponibles en la paleta de Elementos Visuales que se encuentra disponible en la parte derecha del *Editor de Vistas*, tal y como se muestra en la parte inferior derecha de las Figura 2.3a. El objetivo de este tipo de extensión es poder incluir fácilmente en las aplicaciones generadas con EJsS elementos Visuales que un conjunto de usuarios utiliza sistematicamente. Además, da soporte a una inclusión no intrusiva al segundo tipo de modificación soportado por versiones previas de EJsS.
- *Nuevos utilidades en la Barra de Botones*, tal y como se muestra en la parte superior izquierda de la Figura 2.3a. El objetivo de este tipo de extensión es permitir que un usuario con capacidades avanzadas de programación pueda lanzar desde EJsS un conjunto de acciones o aplicaciones que realiza sistemáticamente mientras está utilizando la herramienta.
- *Nuevos Editores en el panel de Edición de Modelos*, tal y como se muestra en la parte superior izquierda de la Figura 2.3b. El objetivo de este tipo de extensión es poder incluir otro tipo de editores, directamente relacionados con el comportamiento/funcionalidad de la aplicación que está desarrollando desde EJsS. Es una alternativa complementaria a los *Elementos del Modelo* que ya existía en EJsS, y que permite a los usuarios destacar directamente en el panel de *Edición de Modelos* algun tipo de funcionalidad que debe utilizar de forma sistemática en sus aplicaciones.

Entendemos que es posible que los tipos de extensiones permitidas, anteriormente mencionadas, no sean las únicas necesarias/útiles para cualquier usuario de EJsS. Aun así, constituyen un conjunto de extensiones lo suficientemente amplio para poder modificar significativamente la funcionalidad y el entorno gráfico de EJsS, y comprobar si la nueva metodología de extensión de EJsS es o no útil. De hecho, en este punto cabe destacar que las creadores/desarrolladores de EJsS están planteándose si las nuevas versiones de EJsS deben dar soporte a la mayor parte de la funcionalidad de la herramienta (incluida la del modelado y simulación de sistemas) bajo este planteamiento.

2.1.5. El uso de los "Plugins" desde EJsS

Para que un nuevo "Plugin" pueda ser utilizado en EJsS, el fichero .JAR que implementa su funcionalidad y elementos gráficos debe estar situado en la carpeta `$WORKSPACE$/config/CusomPlugins/`.



(a) Nuevos elementos en los menús *Editores Principales* y *Vistas*, y en la *Barra de Botones*

(b) Nuevos elementos en el menú *Editores Principales* y en el menú del *Modelo*

Figura 2.3: Opciones de adaptación mediante "Plugins" del interfaz gráfico de EJsS

Esto es así, ya que hemos modificado el proceso de inicialización EJsS para que realice los siguientes pasos:

- *Buscar los "Plugins"* que debe cargar y que se encuentran disponibles en los ficheros .JAR que han sido situados en la carpeta `$WORKSPACE$/config/CusomPlugins/`.
- Para cada fichero .JAR encontrado:
  1. Cargar los recursos (paneles, botones, *Elementos de la Vista* y *Elementos del modelo*).
  2. Añadir los nuevos editores al panel de *Edición Principal*.
  3. Añadir los nuevos botones a la *Barra de Botones*.
  4. Añadir los *Elementos Visuales* al *Editor de la Vista*.
  5. Añadir los nuevos *Editores del Modelo* al panel de *Edición de Modelos*.

Para que este proceso pueda ser llevado a cabo con éxito, en el fichero .JAR además de implementar la funcionalidad y los elementos gráficos (pantallas, menus desplegables, etc) que queremos que sean accesibles desde el entorno gráfico de EJsS, hay que implementar una serie de Interfaces a las clases en las que EJsS espera que se encapsule cada tipo de funcionalidad. Esta forma de proceder define un tipo de arquitectura específica en el software del .JAR que implementa el "Plugin".

2.1.6. La arquitectura e implementación de un "Plugin"

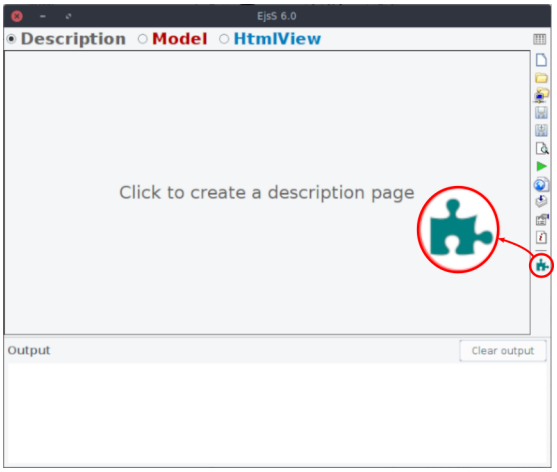
Un "Plugin" se implementa mediante código JAVA y Javascript, que sigue una estructura predefinida que permite a EjsS cargarlo, instanciarlo y comunicarse con él. De esa forma, EJsS reconoce que el fichero .JAR constituye un "Plugin" y puede hacer uso de la funcionalidad y herramientas gráficas proporcionadas por su código.

Brevemente, conviene indicar que todo "Plugin" debe tener una clase principal de JAVA que implementa la Interfaz `Plugin` e implementar un conjunto de métodos que son requeridos por EJsS. Aunque este es el único requisito esencial, otras clases e interfaces JAVA proporcionadas por EjsS están disponibles para construir una nueva extensión. Los más utilizados son la clase `Editor`, que proporciona un editor principal o anidado, y la clase `AbstractModelElement`, que proporciona una plantilla para crear nuevos elementos del modelo. Detalles adicionales de este proceso pueden encontrarse en [9].

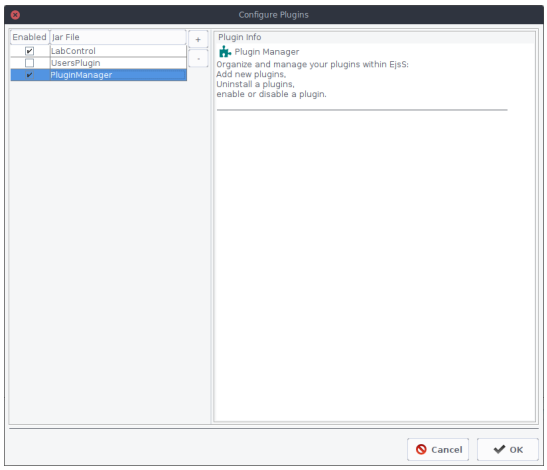
2.1.7. Un ejemplo de "Plugin": el "Gestor de Plugins"

A continuación, se presenta, a modo de ejemplo y de forma breve, un "Plugin" que hemos desarrollado para ilustrar el proceso y para facilitar el proceso de instalación y desistalación del resto de "Plugins". Como esta tarea es una acción que cualquier usuario podría realizar de forma habitual, de forma semejante a las acciones de ejecución o despliegue de modelos, hemos decidido que sea accesible a través de la *Barra de Botones*. Además, como para gestionar los "Plugins" es necesario ver cuales están disponibles y permitir que el usuario instale otro nuevo, es conveniente crear una Ventana de Dialogo, que se abre cuando el usuario presiona el botón correspondiente, desde las que realizar todas estas acciones.

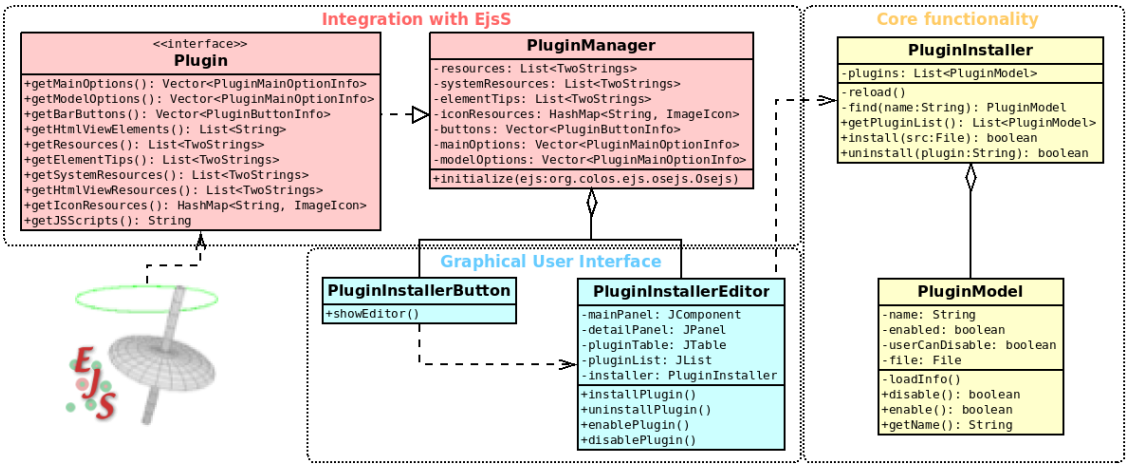
Para ello, se ha desarrollado un "Plugin", que tal y como muestra la Figura 2.4a añade un nuevo botón en EJsS, con el que se despliega la ventana de dialogo que se muestra en la Figura 2.4b. Además, en la Figura 2.4c se muestra el diagrama UML de las clases con las que se implementa el "Plugin": la interfaz `Plugin` y la clase `PluginManager` son necesarias/utilizadas para que EJsS cargue y utilice la funcionalidad del "Plugin"; las clases `PluginInstallerButton` y



(a) Pantalla de EJsS con el nuevo botón



(b) Ventana de diálogo del Plugin



(c) Diagrama UML del gestor de plugins

Figura 2.4: El "Plugin" de "Gestión de Plugins"

`PluginInstallerEditor` implementan la interfaz gráfica del botón y de la ventana de dialogo; y las clases `PluginInstaller` y `PluginModel` implementan la funcionalidad que gestiona los "Plugins".

Ejemplos adicionales de "Plugins" pueden encontrarse en [9] y en la próxima sección, en la que se presenta, desde el punto de vista del usuario, el "Plugin" desarrollado para dar soporte a las tareas de creación, gestión y mantenimiento de nuestro laboratorio remoto de bajo coste.

Finalmente, queremos resaltar que aunque el desarrollo de esta metodología no era un objetivo inicial del PIMCD, constituye una de las contribuciones principales del proyecto, ya que permitirá a otros usuarios adaptar la funcionalidad de EJsS a sus necesidades, y a nosotros seguir mejorando y reutilizar el "Plugin" desarrollado para el laboratorio remoto en versiones futuras de EJsS.

2.2. La gestión del laboratorio remoto desde EJsS

En esta sección se presenta el "Plugin" desarrollado para gestionar de forma centralizada, desde EJsS, un laboratorio remoto de bajo coste que se despliega sobre una Raspberry PI. Para

poder entender mejor el funcionamiento del "Plugin" empezaremos describiendo, de forma somera, el hardware y el software que constituyen el laboratorio remoto, continuaremos explicando las modificaciones que han sido realizados en el software del laboratorio, y terminaremos describiendo la funcionalidad más relevante del "Plugin" desarrollado para su gestión.

2.2.1. El laboratorio remoto original

El laboratorio remoto original, que es mejorado y cuya función se centraliza en EJsS en este PIMCD, fue desarrollado en 2016 y presentado en la Conferencia internacional de Advances in Control Education [6]. En esta sección se presentan los elementos que lo constituyen y las dificultades que los profesores encuentran para su despliegue.

Elementos software y hardware del laboratorio

El laboratorio original utiliza:

- Una Raspberry PI para ejecutar el Código que interactúa, a través de sus pines de entrada y salida, con el sistema real que se desea manipular desde los equipos de los alumnos
- Un servidor de páginas web desarrollado en Node.js para: gestionar el acceso al laboratorio, mostrar la página web que implementa el entorno gráfico mediante el que se manipula el sistema real, almacenar los datos de las experiencia de los alumnos, y mostrar información de ayuda sobre la experiencia que se debe realizar.
- EJsS para definir y generar la página web que contiene el entorno gráfico con el que el alumno interactúa sobre el sistema con el que se desea interactuar.

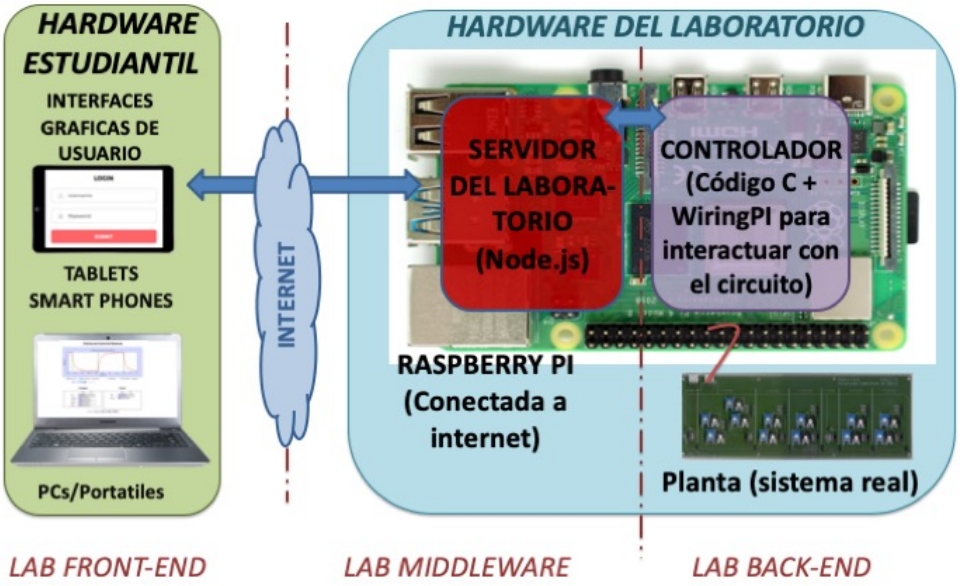
En la Figura 2.5 se esquematiza la relación que existe entre todos estos elementos, que permiten desplegar, con un gasto limitado (una Raspberry PI cuesta menos de 100€) el código con controla un dispositivo real y una página web de un laboratorio remoto, accesible desde los PCs y los dispositivos inteligentes de los alumnos. Es decir, conectando la Raspberry a Internet con una IP publica, y desplegando sobre ella el software desarrollado para una práctica concreta, se dispone de un laboratorio remoto operativo sin necesita de utilizar LMS (Learning Management Systems) más complejos (como por ejemplo Moodle), que no se encuentran disponibles en todas las instituciones de enseñanza.

Problemáticas del despliegue del laboratorio original

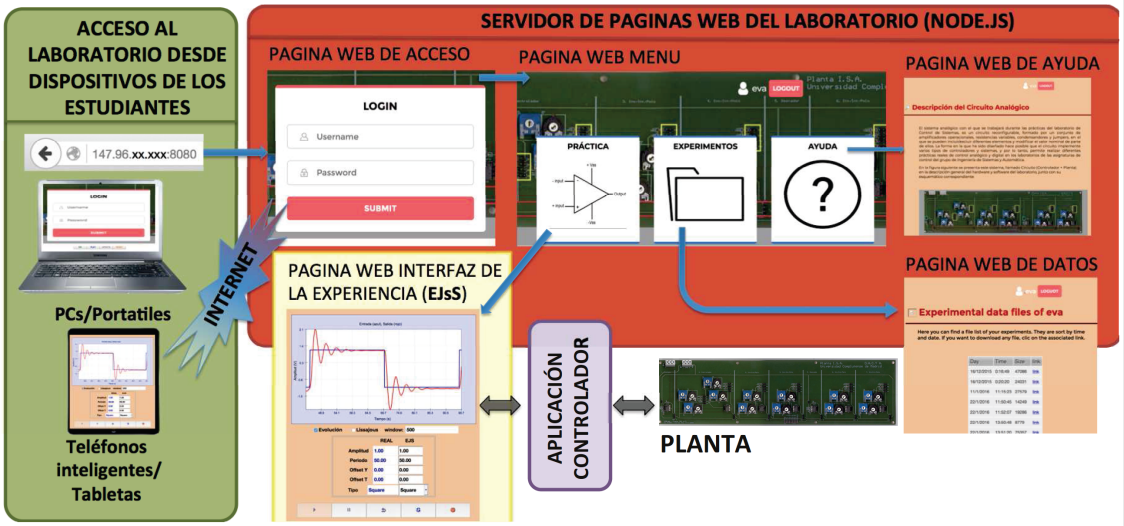
Aunque la metodología presentada en [6] y utilizada en algunas experiencias de las asignaturas de la titulación de Ingeniería Electrónica en Comunicaciones de la Universidad Complutense de Madrid permite poner en marcha un laboratorio remoto operativo de bajo coste, hace necesario que el personal encargado de preparar una nueva experiencia tenga que re-programar, de forma manual, los diferentes elementos software del laboratorio (controlador + servidor + la interfaz gráfica del laboratorio). Para ello, requiere conocer diferentes lenguajes y herramientas (el lenguaje de programación del Controlador, JavaScript, Node.js y EJsS).

Además, las modificaciones relacionadas con el intercambio de información (variables de configuración y de evolución de las señales) entre la interfaz gráfica de la experiencia y el código de Control deben ser realizadas congruente en el código de Control, en el Servidor y en la interfaz gráfica que se desarrolla en EJsS.

Por otra parte, el despliegue de la interfaz gráfica requiere la manipulación manual del código generado por EJsS y su descarga, a través de una aplicación de transferencia de archivos, en



(a) Relación entre los elementos hardware y software



(b) Relación entre las páginas web del laboratorio y el resto de los elementos

Figura 2.5: Relación entre los elementos del laboratorio original

diferentes carpetas de la Raspberry PI. De forma similar, el responsable de la experiencia debe modificar de forma manual y descargar en la Raspberry PI el fichero de usuarios y la página de ayuda del laboratorio. También cada vez que es necesario modificar el código del controlador hay que compilarlo en la Raspberry PI, y por lo tanto, acceder a ella a través de una consola de comandos segura.

Resumiendo, la metodología requiere realizar numerosas labores que involucran diferentes herramientas software y que dificultan que puedan ser llevadas a cabo por un docente con una experiencia

de programación limitada. Para facilitar su labor, hemos decidido centralizar parte de estas laborales desde una versión de EJsS que hace uso de un "Plugin" especialmente desarrollado para nuestro laboratorio. Además, en el proceso de automatización de las labores, el servidor y el código de control también han sido mejorados, de forma que el responsable del laboratorio no tiene que modificar todos los elementos software de forma manual, sino que únicamente tiene que realizar las modificaciones correspondientes a la experiencia concreta que debe realizar en el código de control y configurar un conjunto de parámetros del servidor.

2.2.2. El laboratorio remoto mejorado

En está sección introducimos los cambios más relevantes que se han realizado en la parte del servidor y del controlador, ya que los relacionados con EJsS serán explicados en la sección 2.2.3.

El nuevo servidor del laboratorio

El código del servidor del laboratorio ha sido modificado y re-estructurado con los siguientes objetivos:

- Separar la funcionalidad general de la pagina web del laboratorio (i.e. decir aquella que esperamos que se utilice en todas las experiencias remotas desarrolladas con esta metodología) de los elementos que se deben configurar en cada caso (es decir, aquellos parámetros que hay que proporcionar al laboratorio para que pueda ser desplegado en diferentes Raspberry PIs y para que su comportamiento se adapte a la experiencia concreta para la que ha sido diseñado).

  Para lograr este objetivo, el código responsable de la configuración se encuentra recogido varios ficheros JavaScript, que deben ser editados y modificados por el responsable de la experiencia, y que son accesible a través del EJsS adaptado para la gestión del laboratorio.

- Incluir la funcionalidad necesaria para que el servidor pueda dar respuesta a las acciones que se le solicitan desde EJsS.

  A este respecto conviene destacar que en la versión previa del laboratorio, EJsS únicamente se utilizaba para desarrollar el interfaz gráfico de la experiencia mientras que en la nueva versión, tal y como se detallará en la sección 2.2.3, también es utilizado para gestionar los usuarios del laboratorio, modificar y compilar el código de Control, actualizar el fichero de configuración de EJsS, etc. Por lo tanto, el nuevo servidor debe también atender a las diferentes peticiones, relacionadas con estas nuevas funcionalidades, que se realizan desde EJsS.

- Permitir comprobar fácilmente que el interfaz gráfico de la experiencia funciona correctamente y tiene el aspecto deseado antes de ser desplegado sobre la Raspberry PI.

  La idea subyacente a esta mejora es la siguiente. Para acceder al interfaz gráfico que se ha desplegado a la Raspberry PI el usario debe logearse en el servido de la página web y a continuación seleccionar la página web en la que se implementa dicho interfaz. Una forma alternativa de proceder, que permite simplificar el proceso de pruebas del interfaz, es ejecutar directamente el interfaz gráfico desde EJsS (de forma semejante a la que se ejecuta cualquier otra aplicación desarrollada con la herramienta).

- Modernizar el aspecto de la página web del laboratorio remoto.

  Este cambio es menos relevante, ya que es un retoque estético, sin el que se podría seguir funcionando, pero que genera en los usuarios una experiencia más actual en su interacción con el laboratorio remoto.

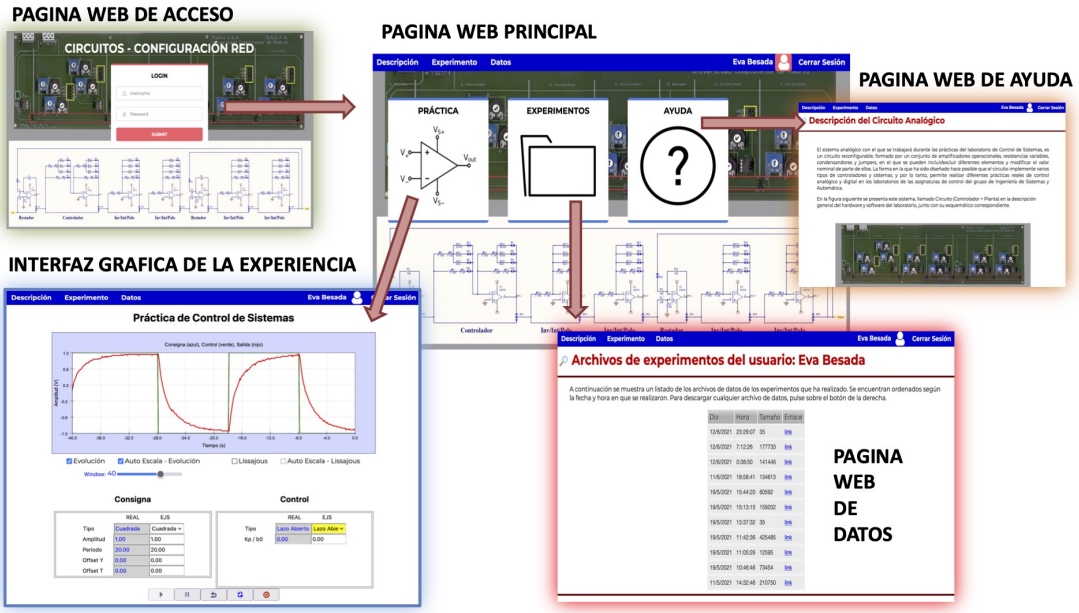


Figura 2.6: Página web del nuevo servidor del laboratorio remoto

En la Figura 2.6 se muestra el aspecto de la nueva página web del laboratorio, para una experiencia análoga a la que se ha utilizado para representar la relación entre las páginas web del laboratorio de la Figura 2.5b. La similitud del aspecto de las páginas web desplegadas por ambas no debe confundir al lector, ya que en ellas no es posible observar directamente la mayoria de las modificaciones anteriormente descritas.

El nuevo código de Control del laboratorio

El código de control de laboratorio que se despliega sobre las Raspberry PI puede escribirse:

- En el **lenguaje de programación C**, de forma análoga a la que se hacía en la versión exterior. Además, en este caso, el programa de Control utiliza su entrada y salida éstandar (STDIO) como método de comunicación con el servidor, y una hebra secundaria para monitorizar si el servido le envía mensajes de reconfiguración y para interpretarlos correctamente. Por otra parte, la hebra principal del programa de control es la que implementa la funcionalidad del programa: lee las señales de salida del sistema real y aplica sobre éste las señales de entrada que considera oportunas, y en caso necesario, entre la lectura y aplicación de señales sobre el sistema real, realiza los cálculos necesarios (p.e. implementa el código de un controlador).

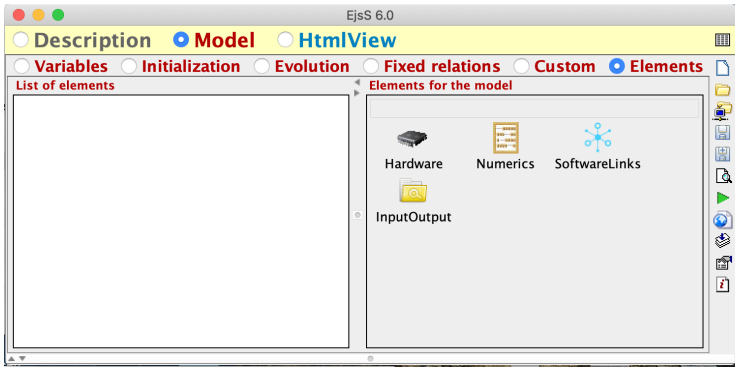
  Sin embargo, como el programa de Control en C depende de la aplicación final del laboratorio, este puede estructurarse en varios ficheros diferentes, que serán compilables de acuerdo con un makefile. Tanto los ficheros en C como el makefile son accesibles y modificables desde EJsS, tal y como se verá en la sección 2.2.3.

- En el **lenguaje de programacion Python**, que no era soportado por la versión previa del laboratorio remoto. Aunque en este caso se podría seguir una metodología similar a la del

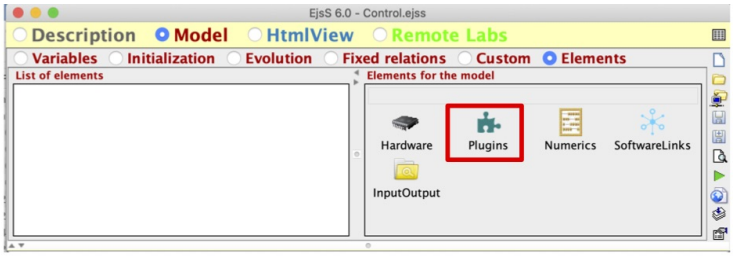
caso anterior, se está siguiendo un procedimiento diferente, ya que de momento, únicamente es utilizado por el brazo robótico, y por lo tanto, hemos decidido desarrollar un protocolo de comunicación específico entre el servidor y el código Python del brazo robótico.

2.2.3. El "Plugin" de EjsS para el laboratorio remoto

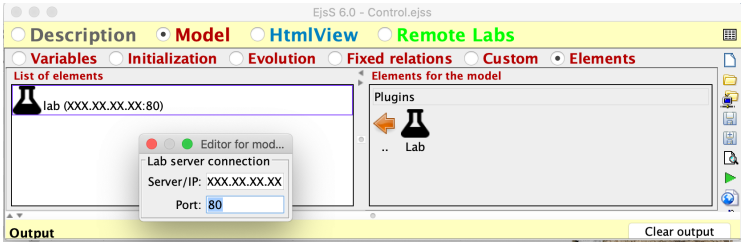
Más allá de las mejoras presentadas en la sección anterior, se encuentran las que se introducen mediante un "Plugin", especialmente desarrollada para facilitar/centralizar la gestión del laboratorio remoto desde EJsS. En esta sección describimos la funcionalidad que se añade a EJsS y los cambios que se producen en su interfaz gráfica (observables en las Figuras 2.7 y 2.8) debido al "Plugin". Además, conviene indicar que en el apéndice B se detalla el proceso seguido para gestionar un laboratorio concreto desde EJsS. y que en [10] se presenta una primera versión del "Plugin" del laboratorio remoto, que ha sido mejorado durante este PIMCD hasta llegar a la situación que se describe a continuación.



(a) Pantalla original de elementos de modelo



(b) Pantalla modificada de elementos de modelo



(c) Configuración del elemento del modelo `Lab`

Figura 2.7: Comparativa entre las pantallas original y modificada de los elementos del modelo

Nuevos elementos de modelo

Para gestionar la información que la aplicación desarrollada en EJsS intercambia con la aplicación de Control, se ha creado una elemento de modelo que está disponible dentro de un nuevo grupo de elementos de modelo "Plugins", no existente en la Figura 2.7a y remarcado en rojo en la Figura 2.7b. Por medio de ese Elemento, se puede crear una instancia al laboratorio remoto tal y como se muestra en la Figura 2.7c, definiendo únicamente su dirección IP y el puerto por el que se establece la comunicación con el servidor. La aplicación desarrollada en EJsS utilizará esa instancia del laboratorio para intercambiar, a través del servidor, toda la información necesaria con la aplicación de control. Además, también es un elemento fundamental para la gestión centralizada del laboratorio desde EJsS, tal y como veremos más adelante, y por lo tanto resulta crucial para poder alcanzar el objetivo **O1**.

Nuevos elementos en la barra de botones

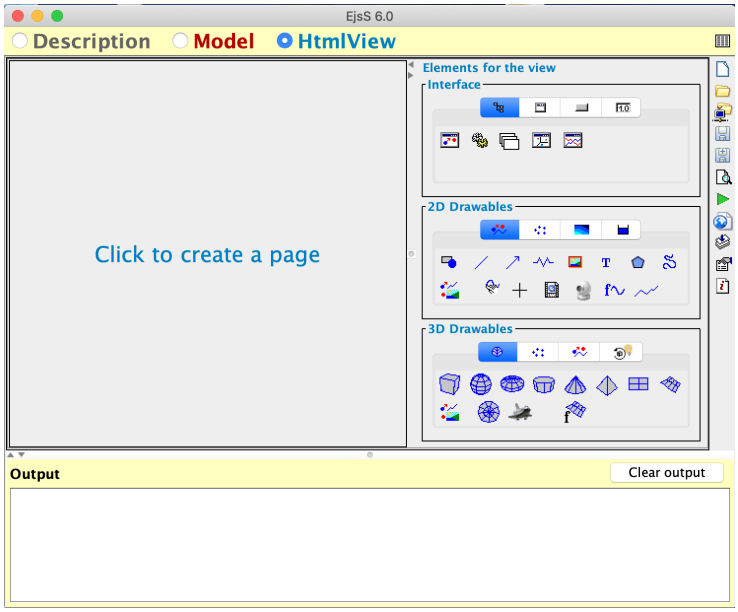
Si comparamos la barra de botones del EJsS original (situada en la parte derecha de la Figura 2.8a) con la barra de botones del EJsS modificado (situada en la parte derecha de la Figura 2.8b), observamos que aparecen dos nuevos botones (enmarcados en azul para facilitar su identificación):

- El primer botón, con forma de puzzle, está asociado al "Plugin" de "Gestión de Plugins", cuya funcionalidad ya ha sido explicada en la sección 2.4.
- El segundo botón está asociado al "Plugin" desarrollado para gestionar nuestro laboratorio remoto, y tiene como primer objetivo la descarga de la interfaz de la experiencia que ha sido desarrollada en EJsS sobre el servidor del laboratorio remoto. Además, durante el proceso de descarga anterior, también descarga la descripción de la experiencia que hay en EJsS sobre el servidor, para que sea utilizada como la página de ayuda/descripción a la que acceden los alumnos desde la página web del laboratorio. Por lo tanto, cumple con los objetivos **O1.C** y **O1.D** de este PIMCD.

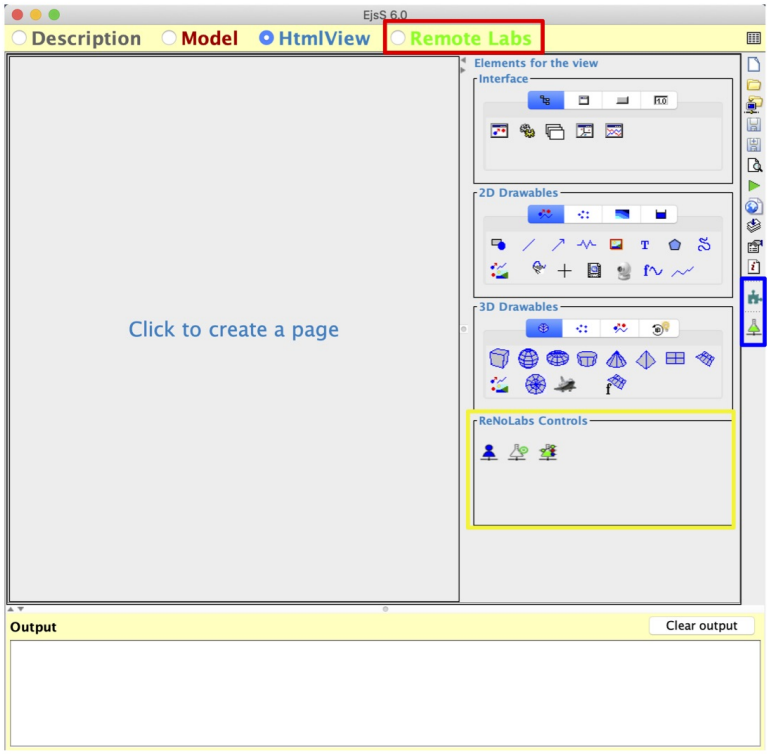
Nuevos elementos en la paleta de elementos visuales

Si comparamos la paleta de elementos visuales del EJsS original (situada en la parte central-derecha de la Figura 2.8a) con la paleta de elementos visuales del EJsS modificado (situada en la parte central-derecha de la Figura 2.8b), observamos que en su parte inferior aparecen tres nuevos elementos (enmarcados en amarillo para facilitar su identificación):

- El elemento visual de la derecha ( ) sirve para generar un panel de parámetros de la experiencia, cuya información se rellena, de forma automática, a partir de los datos que intercambia el servidor con el interfaz gráfico de la experiencia al inicio de la misma. Por lo tanto, está relacionado con el objetivo **O1.A** de este PIMCD, con el que se busca automatizar la conexión de variables entre los diferentes elementos software del laboratorio.
- El elemento visual central ( ) sirve para generar el panel de control (mediante el que se inicializa, para, resetea o actualizan los parámetros de la experiencia) del interfaz grafico del laboratorio remoto. Por lo tanto, también se encuentra relacionado con el objetivo **O1.A** de este PIMCD.
- El elemento visual de la izquierda ( ) sirve para incluir en el interfaz gráfico de la experiencia un panel desde el que hacer el login de un usuario. Esta funcionalidad, que inicialmente no se encontraba contemplada entre los objetivos del PIMCD, tiene como objeto facilitar



(a) Pantalla original de EJsS



(b) Pantalla de EJsS tras cargar el “Plugin”

Figura 2.8: Comparativa entre la pantalla original y modificada de EJsS

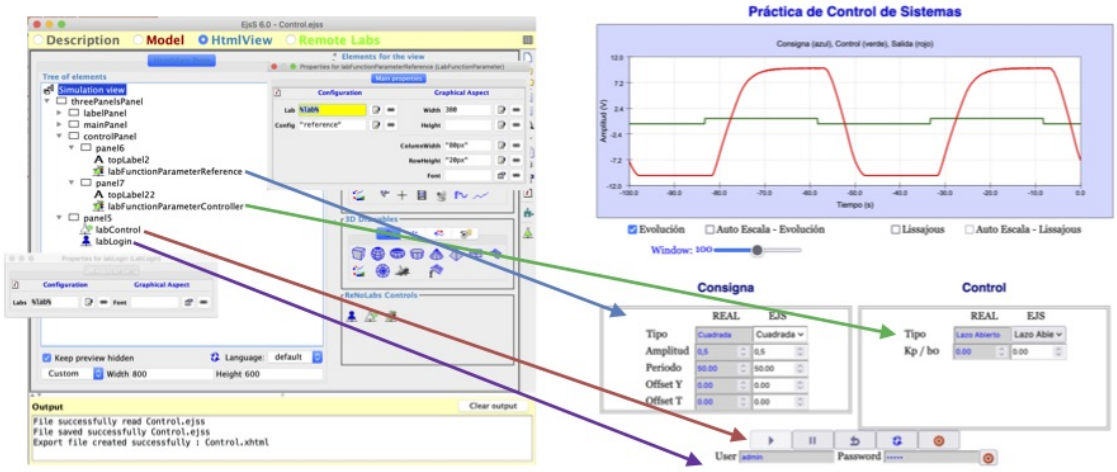


Figura 2.9: Ejemplo de uso de los elemento gráficos del "Plugin" en una experiencia

la depuración del entorno gráfico y permitir el acceso a la experiencia desde la aplicación generada desde EJsS sin que el usuario tenga que acceder a la página web del laboratorio remoto.

En la Figura 2.9 mostramos sobre el interfaz gráfico de una de nuestras experiencias de laboratorio, los tres elementos gráficos anteriores. En la configuración de cada uno de ellos hemos tenido que indicar la instancia del elemento de modelo `Lab` que estamos utilizando (de forma que cuando se ejecuta cualquier acción con estos elementos visuales, la aplicación interactúa correctamente con el servidor del laboratorio elegido).

Finalmente, cabe indicar que el elemento relacionado con el panel de login no está visible de forma habitual en la interfaz gráfica de nuestras experiencias, ya que lo deshabilitamos una vez que la práctica está totalmente desarrollada porque preferimos que los alumnos accedan a las experiencias a través de la página web del laboratorio remoto.

Nuevo elemento en el menú de Edición principal

Si comparamos el menú de Edición Principal del EJsS original (situado en la parte superior de la Figura 2.8a) con el menú de Edición Principal del EJsS modificado (situado en la parte superior de la Figura 2.8b), observamos que aparece un nuevo elemento (enmarcado en rojo para facilitar su identificación). Si se selecciona el nuevo Editor se observa la pantalla que aparece en la Figura 2.10, mediante la cual, se puede elegir cada una de las posibilidades que se ofertan en el menú remarcado en verde en dicha imagen. En concreto, a través de este editor principal es posible:

- Configurar y comprobar la conexión con el laboratorio remoto. Para ello, hay que cargar los elementos `Lab` existentes en el Modelo y seleccionando uno de ellos. A continuación, hay que indicar el nombre y password del usuario, y solicitar que se establezca la conexión con el laboratorio remoto. Esta paso es fundamental para poder llevar a cabo las siguientes labores de gestión del laboratorio desde EJsS.

- Obtener, modificar y descargar los ficheros de configuración del servidor del laboratorio, a través del submenú de configuracion que se presenta en la Figura 2.11. Como ya se ha mencionado anteriormente, hay una pequeña parte de la funcionalidad del laboratorio que

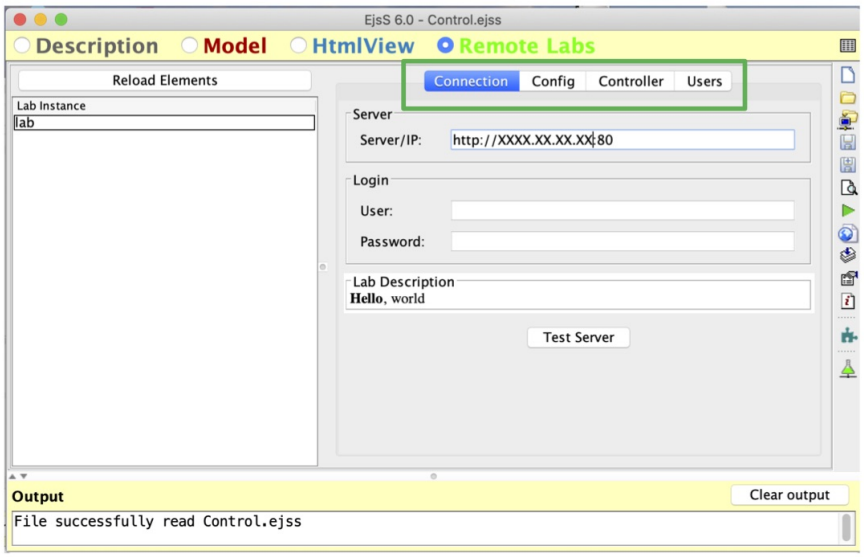


Figura 2.10: Menú de edición del laboratorio remoto con el submenú de conexión elegido

debe ser adaptada a cada experiencia y que se encuentra disponible en varios fichero de configuración. En el que se muestra en la Figura 2.11a se configuran los parámetros generales del laboratorio y en el de la Figura 2.11b los parámetros específicos de control de cada experimento y las variables que hay que almacenar en cada caso. Por lo tanto, a través de esta opción, se centraliza desde EJsS la configuración de la funcionalidad del laboratorio relacionada con los objetivos **O1.A** y **O1.E** de este PIMCD.

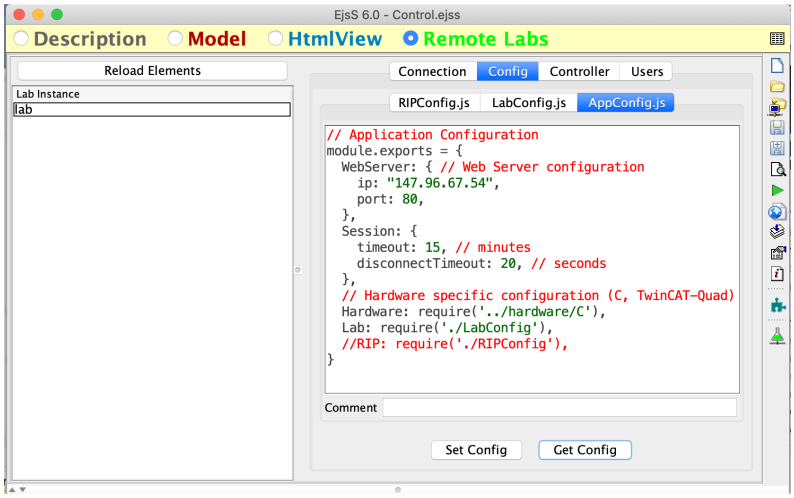
- Obtener, modificar, descargar y compilar los ficheros en los que se define el comportamiento del código del programa Controlador, desde el que se interactúa directamente con el sistema real y que intercambia información con el interfaz de la experiencia de forma indirecta, a través de una pasarela establecida por servidor del laboratorio. Además, también permite cargar, modificar y descargar el `makefile` utilizado para compilar la aplicación del controlador. Por lo tanto, la funcionalidad correspondiente a este caso está directamente relacionada con el objetivo **O1.B** del PIMCD. Para realizar todas estas labores, se utiliza el submenú del controlador que aparece en la Figura 2.12.

- Obtener, modificar y descargar la información del fichero que almacena los datos de todos los usuarios que pueden acceder a la experiencia del laboratorio, a través del submenú que aparece en la Figura 2.13. Por lo tanto, la funcionalidad correspondiente a este caso está directamente relacionada con el objetivo **O1.F** del PIMCD.

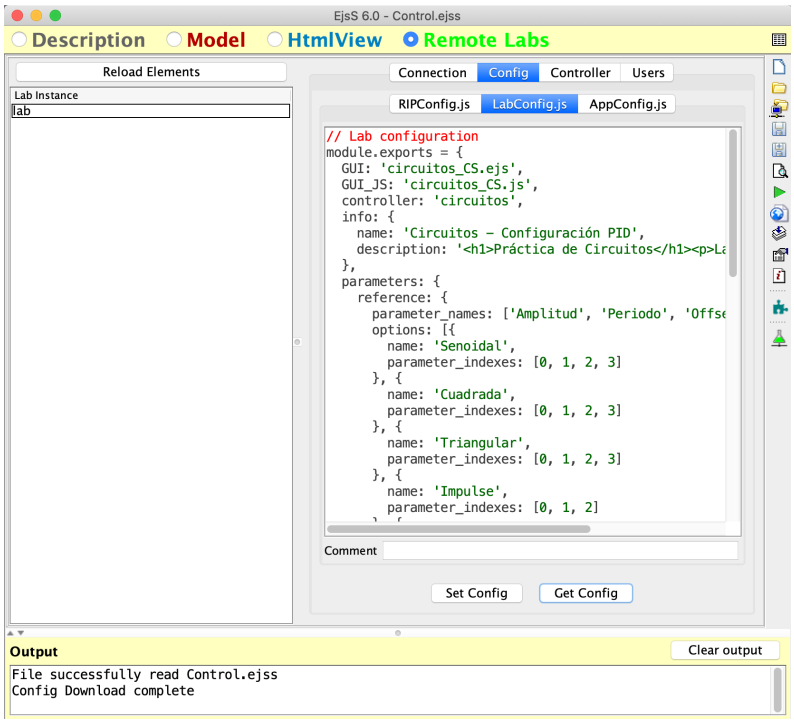
2.3. Documentación disponible

La documentación que se ha desarrollado para los nuevos usuarios de la metodología y herramientas que se plantea en este PIMCD para el desarrollo de laboratorio remotos de bajo coste es la siguiente:

- Un manual de instalación de las diferentes herramientas software que se necesitan para poder seguir la metodología. El contenido de este manual constituye el apéndice A y se encuentra



(a) Configuración del los parámetros generales del servidor



(b) Configuración del los parámetros específicos del controlador

Figura 2.11: Menú de edición del laboratorio remoto con el submenú del configurador del servidor elegido

también disponible en forma de vídeo, en el repositorio creado para este proyecto [11].

- Un manual de usuario del desarrollo, la puesta a punto y gestión los laboratorio remotos desde EJsS. El contenido de este manual constituye el apéndice B y se encuentra también disponible en forma de vídeo, en el repositorio creado para este proyecto [11].

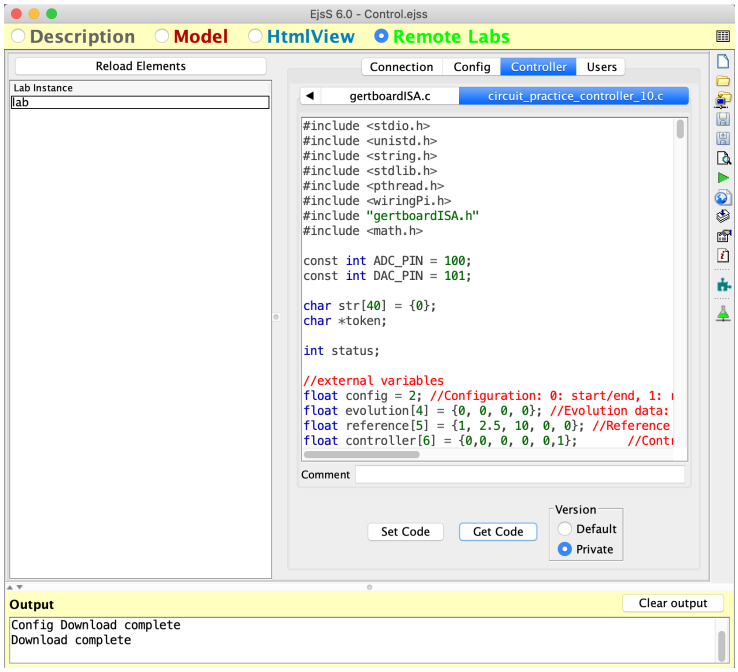


Figura 2.12: Menú de edición del laboratorio remoto con el submenú del controlador elegido

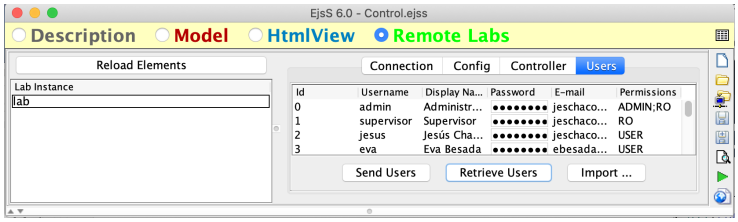


Figura 2.13: Menú de edición del laboratorio remoto con el submenú de los usuarios elegido

Además, tanto la memoria de este proyecto PIMCD, que contiene información detallada de toda la metodología, como las publicaciones que se resumen en la sección 3.5 constituyen un buen punto de partida para tener una vista general de diferentes aspectos de la misma.

Finalmente, conviene destacar que todo esta documentación da respuesta al objetivo **O2.A** de este PIMCD.

Capítulo 3

Uso de la herramienta desarrollada para actualizar y crear nuevas experiencias de laboratorio

Las secciones 3.1 y 3.2 de este capítulo presentan los laboratorios remotos que han sido actualizados o creados, durante el desarrollo de este PIMCD, mediante la metodología y herramientas presentadas en el capítulo anterior. Conviene indicar que estos laboratorios han sido organizados en este capítulo de acuerdo con sistema real con el que interactúan, y no según el proceso seguido a lo largo del proyecto para su actualización o creación. Este es así, ya que consideramos conveniente presentar al lector el sistema real sobre el que se realizará cada tipo de experiencia, antes de detallar cada uno de los laboratorios disponibles sobre dicho sistema.

Además, para poder analizar los laboratorios disponibles desde la perspectiva de los objetivos del proyecto, en la sección 3.3 se presenta un resumen de todos ellos y su estado actual de desarrollo y uso. Por otra parte, la sección 3.4 de este capítulo resume toda la documentación que ya se ha generado para los alumnos, sobre las diferentes experiencias remotas que han podido realizar este curso o que estarán disponibles a partir del próximo. Finalmente, la sección 3.5 resume los procedimientos utilizados para difundir, entre la comunidad educativa, la metodología y las herramientas desarrolladas durante este proyecto PIMCD.

3.1. Laboratorios remotos sobre circuitos analógicos

En esta sección describimos el sistema real reconfigurable sobre el que los alumnos realizan la mayor parte de las prácticas con dispositivos reales de las asignaturas de Sistemas Lineales, Procesamiento de Señales y Control de Sistemas de la titulación de Ingeniería Electrónica en Comunicaciones de la UCM. Además, damos una visión general de la relación que se establece entre los elementos hardware y software que constituyen los laboratorios remotos que involucran a este sistema, y por último, describimos cada una de las experiencias remotas que se han adaptado/creado a/con la nueva metodología.

3.1.1. El sistema real: un circuito analógico reconfigurable

El sistema analógico con el que se trabaja en diferentes experiencias del laboratorio es el circuito reconfigurable mostrado en la parte superior de la Figura 3.1. Está formado por amplificadores operacionales, resistencias variables, condensadores y jumpers, mediante los que se pueden incluir/excluir diferentes elementos del circuito y modificar el valor nominal de alguno de ellos.

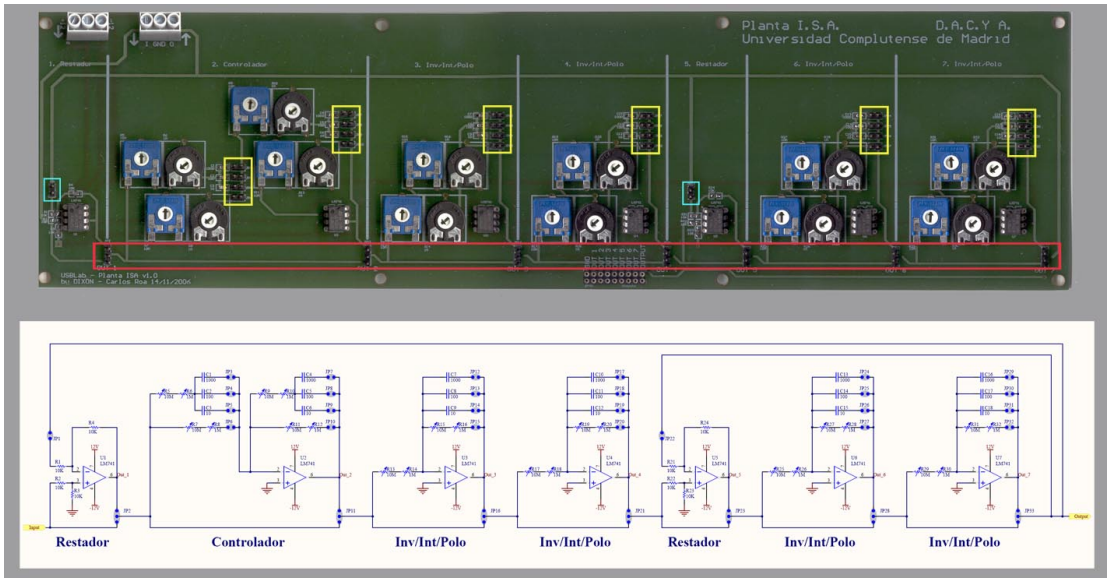


Figura 3.1: Circuito analógico reconfigurable

La forma en la que ha sido diseñado hace posible que el circuito implemente, a través de los subsistemas que se muestran en la parte inferior de la la Figura 3.1, varios tipos de reguladores (redes y PIDs), sistemas lineales que se pueden construir colocando en serie (y realimentando de forma unitaria) ganancias, integradores y polos simples, y la realimentación unitaria de un lazo cerrado básico (en el que en el lazo principal se sitúa en serie un controlador y una planta). Por lo tanto puede ser utilizado para que los alumnos analicen el comportamiento de diferentes Sistemas Lineales, para que construyan algunos filtros básicos de Procesamiento de Señales, y para analizar el comportamiento de diferentes reguladores de Control de Sistemas.

Más concretamente, el circuito está formado por varios elementos básicos (1 Controlador, 2 Restadores, y 4 Inv/Int/Polos) dispuestos en serie, que pueden ser incluidos o excluidos de los experimentos por medio de la fila de jumpers enmarcados en rojo en la figura. Todos los jumpers de esta fila tienen que estar colocados, bien en su posición inferior (caso en el que que excluirá al elemento colocado antes del jumper del circuito utilizado) o en su posición superior (caso en el que se incluirá el elemento, en serie con el resto de los elementos elegidos).

Cada uno de los elementos básicos está formado a su vez por un conjunto de jumpers de configuración (enmarcados en amarillo y azul en la figura) y, opcionalmente, un conjunto de resistencias variables y condensadores de diferentes valores (1000, 100 y 10 nF). Los valores de las resistencias se ajustan mediante dos potenciómetros en serie (que permiten elegir valores en el rango [0,11] MΩ) y el de los condensadores se elige a través del jumper colocado a su derecha.

Para conectar el circuito a cualquier dispositivo, se utilizan las clemas colocadas en su parte superior izquierda: la que está más a a la izquierda sirve para alimentar los amplificadores operacionales del circuito (con ± 12V) y la que está más a la derecha sirve para conectar las señales de entrada y salida al circuito. Además, las valores de salida de cualquiera de los elementos básicos del circuito están disponibles en la ristra de pines que se sitúa en la parte inferior del circuito.

Detalles adicionales de este circuito, entre los que se incluyen las funciones de transferencia de cada elemento básico y como debe ser configurado en cada caso, se presentan en el anexo D.

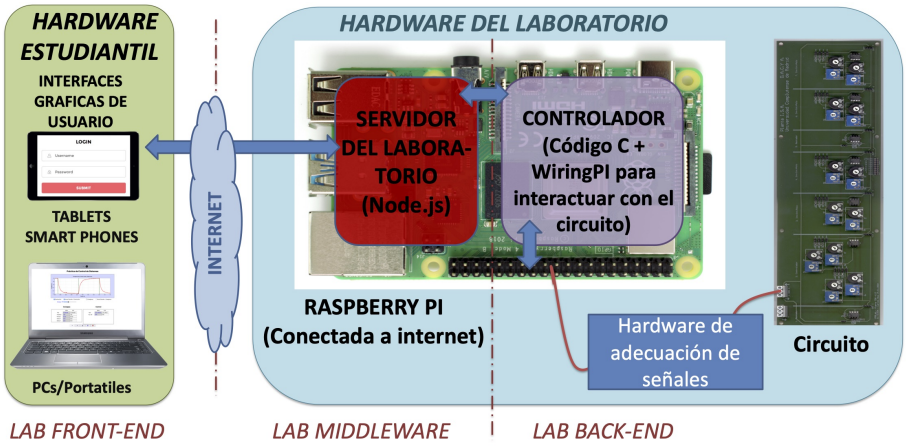


Figura 3.2: Vista general de los elementos involucrados en los laboratorios remotos sobre circuitos

3.1.2. Visión general de las experiencias remotas sobre el circuito analógico

En la Figura 3.2 se esquematiza la relación que existe entre los elementos hardware y software que se utilizan para hacer una experiencia remota sobre el circuito analógico. Por una parte, a la izquierda, encontramos el equipamiento de los alumnos, con el que acceden al laboratorio remoto e interactúan con él a través de internet. En la parte central superior situamos la Raspberry PI, sobre la que se despliega el servidor de la página web del laboratorio remoto (que debe ser reconfigurado mínimamente para cada tipo de experiencia) y el programa de Control (programado en el lenguaje C y que debe ser adaptado a cada experiencia). Finalmente, en la parte inferior y derecha de la imagen se esquematiza el intercambio de información entre el programa de Control y los pines de entrada y salida de la Raspberry PI, a los que se conecta, una etapa hardware para adecuar los valores de las señales de la Raspberry PI a los valores requeridos por las señales de entrada y salida del circuito analógico reconfigurable.

Más concretamente, el hardware de adecuación de señal incluye los siguientes elementos:

- El conversor digital analógico (DAC) MCP4812 de Microchip, responsable de convertir las señales digitales que el programa de Control intercambia con él a través del protocolo SPI en señales analógicas (en concreto tensiones en el rango [0,4.96] V).
- El conversor analógico digital (ADC) MCP3008 de Microchip, responsable de convertir las señales analógicas externas a valores digitales que el programa de Control obtiene de este dispositivo a través del protocolo SPI.
- Dos amplificadores operacionales configurados de forma adecuada (con la ayuda de varias resistencias) para convertir las tensiones generadas por el código del Control, en el rango [0,4.96] V a la salida del DAC, en tensiones de entrada al circuito, con valores válidos en en el rango [-12,12] V.
- Otros amplificadores operacionales configurados de forma adecuada (con la ayudas de varias resistencias) para convertir las tensiones generadas como salidas del circuito, originalmente en el rango [-12,12] V, en tensiones de entrada al ADC, con valores válidos en el rango [0,3.3] V.

Además, el código de control debe a su vez, complementar las operaciones de los amplificadores operacionales, de forma que calcule las señales que debe enviar al DAC en el rango [0,4.96] V a partir de las señales originalmente situadas en el rango [-12,12] V, y que convierta las señales que recibe del ADC en el rango [0,3.3] V a los valores que originalmente generaba la salida del circuito en el rango [-12,12] V. Más allá de esta conversión de las señales que envía al DAC y recibe del ADC, a través los pines de entrada y salida de la Raspberry PI, y por medio de funciones de la librería Wiring PI, el código de Control también debe calcular, al menos, el valor de la señal de referencia (que aplica en cada instante al sistema o al lazo cerrado), e intercambiar con el servidor, y por ende con la interfaz gráfica del laboratorio, los valores de sus parámetros y la evolución de un conjunto de señales del experimento. La funcionalidad particular de cada experiencia será discutida a continuación, en las siguientes secciones.

Finalmente, conviene indicar, para aquellos lectores interesados en profundizar en los detalles del hardware de adecuación, que éste se encuentra descrito en el apéndice E.

3.1.3. Experiencias para la asignatura de Sistemas Lineales

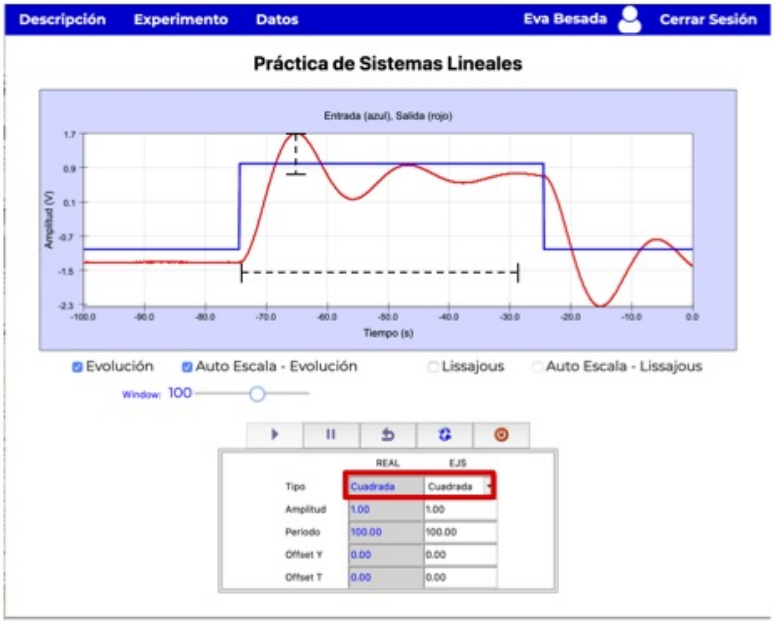
En la asignatura de Sistemas Lineales los alumnos realizan dos tipos de experiencias diferentes sobre el circuito. Ambas tienen como objeto determinar el modelo matemático (en concreto la función de transferencia) asociada a una configuración del circuito que los alumnos conocen solo parcialmente, bien a través del análisis de la respuesta temporal del sistema a una entrada escalón, o del análisis de la respuesta permanente del sistema a sinusoides de diferente frecuencia.

Aunque la configuración del circuito y la señal de entrada (escalón o sinusoide) es diferente en cada experiencia, los alumnos utilizan el mismo interfaz gráfico de la experiencia para tomar los datos de la práctica. Más en concreto, utilizan el interfaz que ha sido diseñado para el laboratorio remoto de Sistemas Lineales, y que se muestra en la Figura 3.3 con configuraciones válidas para ambas experiencias:

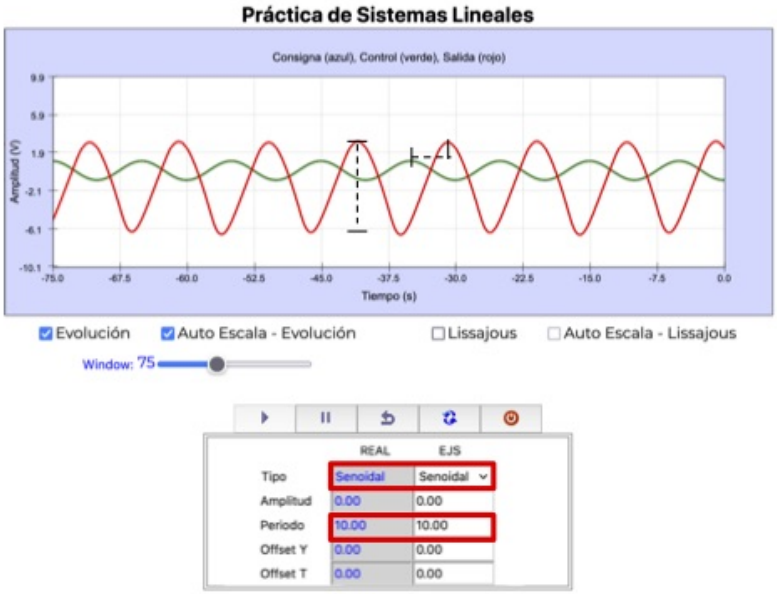
Primera experiencia. Sobre la evolución de la respuesta experimental (curva roja) que observan en una gráfica similar a la de la Figura 3.3a, deben determinar el tiempo que tarda en dejar de oscilar, el pico, etc (i.e. valores asociados a las líneas marcadas en negro), y con estos valores obtener el modelo del sistema.

Segunda experiencia. Sobre el régimen permanente de la respuesta temporal (curva roja) que se observa en la gráfica de la Figura 3.3b, los alumnos deben obtener (cambiando los valores de la frecuencia de la sinusoide) la amplitud y el desfase de la señal de salida (i.e. valores asociados a las líneas marcadas en negro), y a partir de los datos tomados para diferentes frecuencias, representar el diagrama de Bode del sistema, analizarlo, y obtener el modelo que lo describe.

Además, desde el punto de vista del código de control (escrito en C), cabe indicar que éste básicamente tiene que calcular la señal de referencia (azul en las figuras) que le debe aplicar al sistema en cada momento, convertirla al valor entero adecuado para enviársela al DAC, y leer la respuesta del sistema a través del ADC. Todos estos valores son continuamente enviados al servidor de prácticas para que éste se los reenvíe al interfaz gráfico de la experiencia. Además, el controlador también recibe órdenes del servidor, indicándole los nuevos valores con los que generar la referencia y los botones (play, stop, reset o update) pulsados por el alumno para que pueda adecuar su funcionamiento a los mismos (ejecutar el código, parar, resetear los parámetros o actualizarlos con los valores indicados por el usuario en el interfaz gráfico). El funcionamiento del controlador y la forma que éste intercambia información con la interfaz de la experiencia a través del servidor se esquematiza en la Figura 3.4.



(a) Análisis respuesta temporal



(b) Análisis respuesta en frecuencia

Figura 3.3: Interfaz de las experiencias de Sistemas Lineales

Finalmente, cabe indicar que información detallada sobre ambas experiencias se recoge en el guión que se le entrega a los alumnos en cada caso, y que se ha recogido en este documento en los apéndices F y G. Además, la primera experiencia es una adaptación a la nueva metodología de otra

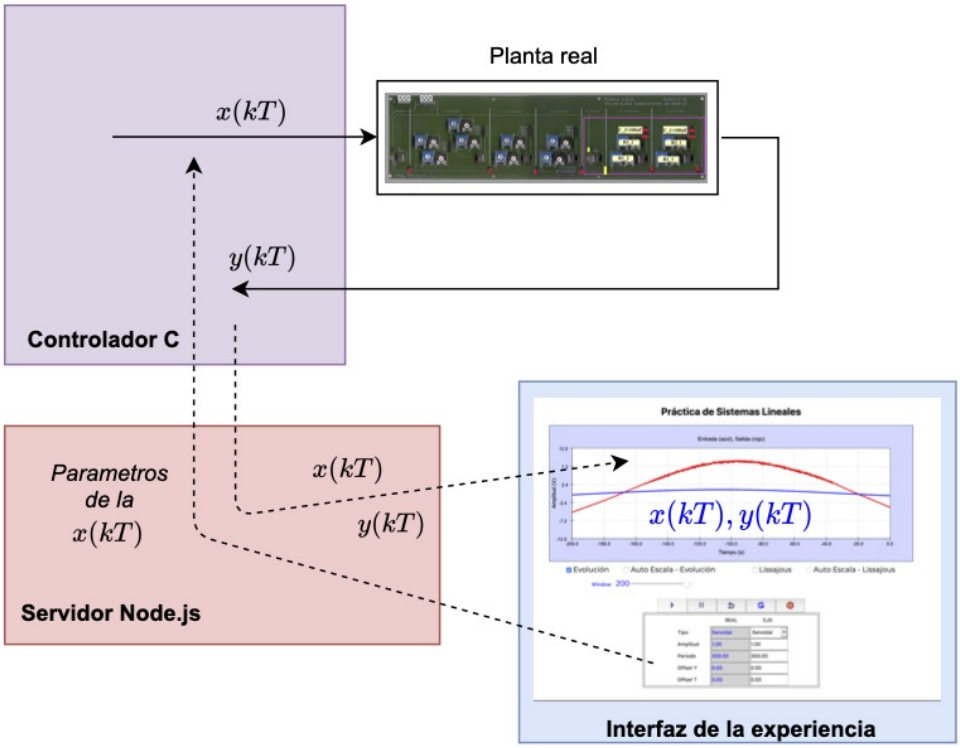


Figura 3.4: Relación entre el código de control, servidor y la interfaz gráfica

que ya estaban realizando de forma remota los alumnos desde hace varios cursos, y cuyos resultados son satisfactorios en un porcentaje elevado de alumnos, motivo por el que los alumnos identifican dos sistemas diferentes, cada uno conectado a una Raspberry PI. Sin embargo, aunque la segunda experiencia también es una adaptación de software disponible, ha sido utilizada por primera vez durante este curso académico para que los alumnos realizasen de forma remota la práctica. El resultado en este caso, no ha resultado adecuado, porque es una práctica un poco larga, en la que los alumnos necesitan el apoyo de los profesores durante más tiempo. Por lo tanto, en el futuro se realizará como herramienta de apoyo a la práctica presencial, para que los alumnos terminen su práctica de forma remota o completen la recogida de datos que necesiten para refinar el diagrama de Bode.

3.1.4. Experiencia para la asignatura de Procesamiento de Señales

Para la asignatura de Procesamiento de Señales se ha preparado en este Proyecto una nueva experiencia, para que identifiquen el modelo matemático del circuito a través del análisis de la respuesta del mismo ante una entrada aleatoria ergódica. Esta experiencia complementa las realizadas previamente en la asignatura de Sistemas Lineales, utilizando los conocimientos adquiridos sobre señales aleatorias en la asignatura de Procesamiento de Señales para obtener el diagrama de Bode experimental de forma eficiente.

Los alumnos utilizan una interfaz de experiencia similar a la de Sistemas Lineales para llevar a cabo esta experiencia. La diferencia principal radica en que, tal y como se puede ver en la Figura 3.5,

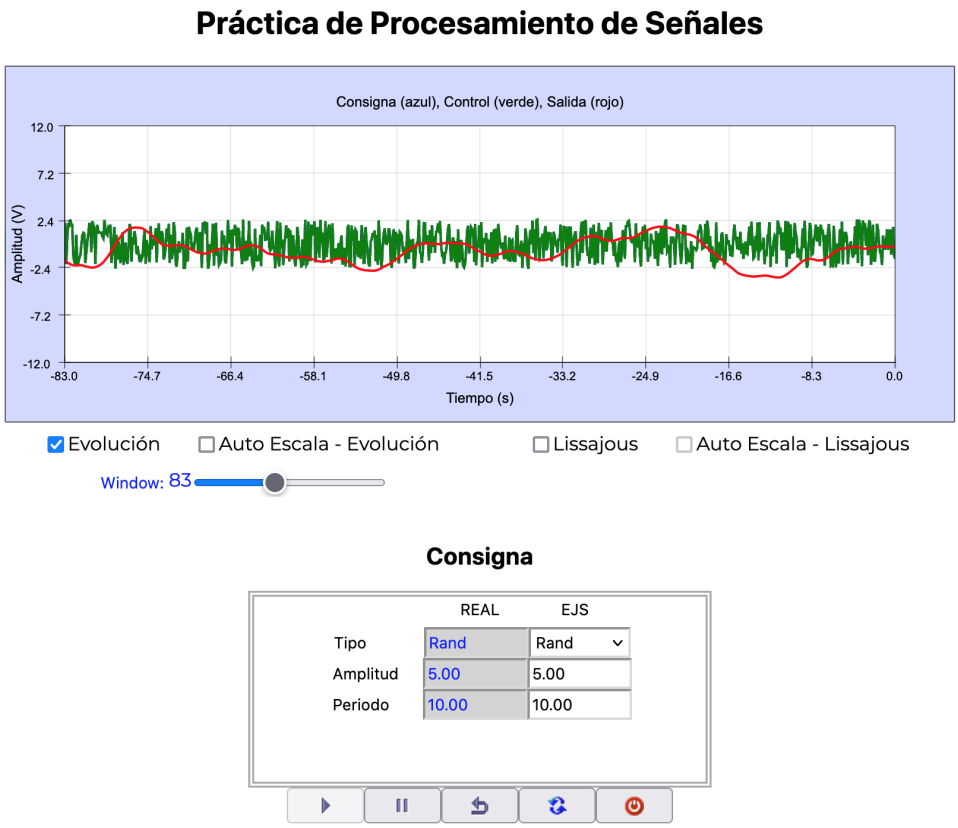


Figura 3.5: Interfaz de las experiencias de Procesamiento de Señales

disponen de nuevas señales en el menú desplegable de señales de referencia, de forma que puedan elegir una señal ergódica entre ellas. A continuación deben descargarse los valores de las señales de entrada y salida en su equipo, para poder procesarlos correctamente (calculando autocorrelaciones, correlaciones cruzadas y transformadas rápidas de Fourier), obtener el diagrama de Bode, y a partir de éste obtener la función de transferencia del sistema configurado en el circuito analógico.

Desde el punto de vista del código de Control, el comportamiento es también similar al del caso anterior. La única diferencia radica en que el código ha sido ampliado para que pueda también generar las señales aleatorias que se pueden elegir a través del menú utilizado para configurar la señal de referencia.

Además, los ficheros de configuración del servidor también han sufrido modificaciones, de forma que ahora incluyen la información necesaria para las nuevas señales.

Con estos cambios, realizados desde la herramienta EJsS mejorada con el "Plugin" desarrollado para el laboratorio remoto, ha sido sencillo poner a la disposición de los alumnos una nueva experiencia, que podrá ser utilizada por los alumnos de Procesamiento de Señales a partir del curso que viene. Detalles adicionales de esta experiencia y de la información/guión preparado para que los alumnos puedan llevarla a cabo, pueden encontrarse en el apéndice H.

3.1.5. Experiencias para la asignatura de Control de Sistemas

En la asignatura de Control de Sistemas, los alumnos realizaban, en el laboratorio presencial, diferentes experimentos de diseño de controladores sobre el circuito real, que han sido realizados este curso, gracias a la nueva versión de la herramienta, parcialmente de forma remota. Además, la parte de la experiencia presencial, que es necesaria ya que los alumnos tienen que modificar los valores de los condensadores y resistencias de parte del circuito, ha sido realizada también con la misma herramienta.

Más en concreto, los alumnos realizan dos experiencias diferentes:

1. *Primera experiencia:* Diseño de una red de adelanto para modificar la respuesta temporal y en frecuencia de un sistema conocido.

   Para esta experiencia se les proporciona a los alumnos el modelo matemático del circuito que se ha configurado sobre la planta. A continuación los alumnos deben diseñar, sobre dicho modelo, una red de adelanto que permita que el comportamiento del sistema en lazo abierto y en lazo cerrado tengas unas características determinadas. Finalmente, es necesario que prueben los controladores diseñados sobre el sistema real, momento en el que deben interactuar con él a través de las herramientas actualizadas/mejoradas con la nueva metodología.

2. *Segunda experiencia:* Diseño de diferentes controladores de la familia PID (Proporcional, Integral y Derivativo) para modificar la respuesta temporal de un sistema desconocido.

   En este caso, a los alumnos se les proporciona un modelo caja negra (que pueden simular, pero del que no saben las expresiones matemáticas) del circuito sobre el que van a trabajar, de forma que puedan diseñar y probar el funcionamiento de diferentes controladores de la familia PID sobre él. Después de esta primera etapa de diseño y análisis, deben probar el funcionamiento de los controladores diseñados sobre el circuito real, siendo necesario en este punto las herramientas actualizadas/mejoradas con la nueva metodología.

En ambas experiencias, las pruebas de los controladores se realizan de dos forma diferentes:

- *Estudio del funcionamiento de las variantes discretas del controlador.*

  En este caso el alumno debe obtener la función de transferencia discreta del controlador, introducir sus parámetros en el interfaz del laboratorio, que se muestra en la Figura 3.6 y que ha sido diseñado para las experiencias de Control de Sistemas, y observar en las curvas de evolución la respuesta del sistema controlado.

  Esta parte de la experiencia se puede realizar de forma remota, ya que el controlador se configura a través de la interfaz de la experiencia.

- *Estudio del funcionamiento de las variantes continuas del controlador.*

  En este caso el alumno tiene que obtener los valores de las resistencias y condensadores de los elementos del circuito que sirven para implementar el controlador continuo correspondiente y configurar el circuito de forma adecuada. A continuación, utilizará la interfaz gráfica de la experiencia para que genere la señal de referencia de entrada al lazo cerrado y registre su respuesta.

  Esta parte de la experiencia debe realizarse de forma presencial, para que los alumnos puedan manipular los elementos necesarios del circuito reconfigurable para probar los controladores continuos que han obtenido.

Desde el punto de vista del código del controlador, este incorpora dos formas de funcionar diferentes, seleccionables a través de la interfaz por el tipo de control elegido (discreto vs. continuo).

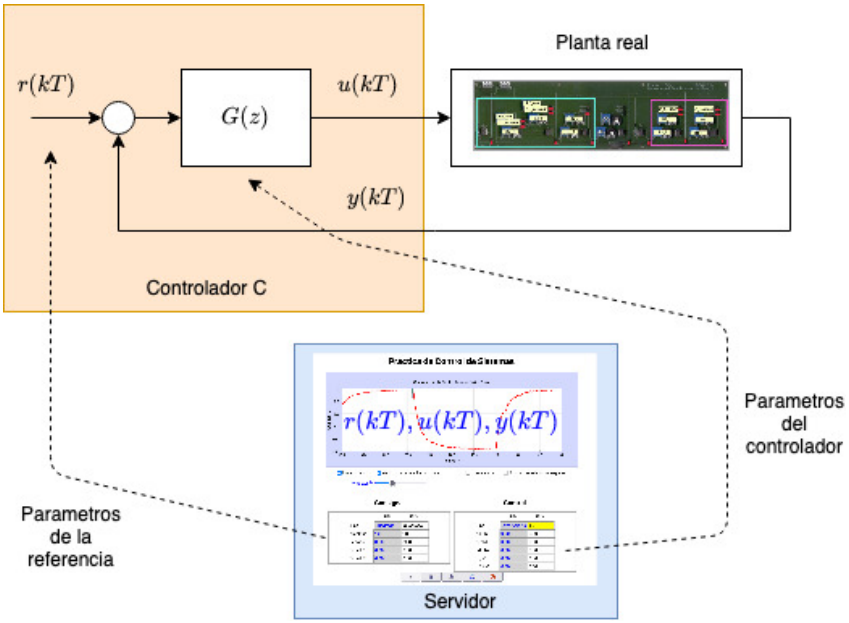
Figura 3.6: Interfaz de las experiencias de Control de Sistemas

- *Control discreto.* El código del controlador, cierra el lazo de control sobre la planta, tal y como se esquematiza sobre la Figura 3.7a, a través de una etapa de: medición de la respuesta del circuito, cálculo de la referencia, obtención de la diferencia entre la respuesta y la referencia, y cómputo de la señal de control que se aplica sobre la entrada de la planta teniendo en cuenta la ecuación en diferencias asociada el modelo matemático del controlador discreto que está siendo aplicado.

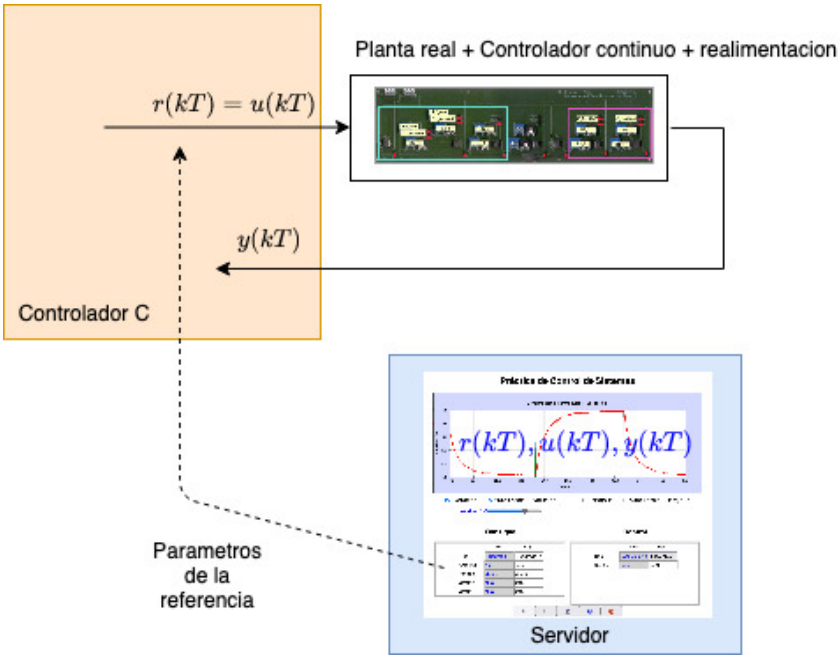
- *Control continuo.* En este caso, el lazo de control se cierra directamente sobre el circuito reconfigurable y el código de control funciona, tal y como se muestra en la Figura 3.7b, de forma similar a la de los casos de las prácticas de Sistemas Lineales y Procesamiento de Señales: calcula la señal de referencia, la aplica sobre la planta, y recoge su respuesta.

Desde el punto de vista del servidor, es necesario incluir en sus parámetros de configuración las variables que son utilizadas para configurar el tipo de control (discreto vs. continuo) y los parámetros de los controladores discretos. El interfaz gráfico desarrollado en EJsS para esta experiencia, que se puede ver en la Figura 3.6, también incorpora un menú desplegable adicional, que permite elegir y parametrizar el comportamiento de los diferentes controladores.

Estas prácticas han sido realizadas, siguiendo la metodología remota para la mayor de la experiencia (incluida la de los controladores discretos) y la presencial para los controladores continuos, por primera vez durante este curso académico. Los resultados, podemos considerar, que han sido



(a) Control discreto



(b) Control continuo

Figura 3.7: Diferencias entre los experimentos con control discreto y continuo

análogos a los de años anteriores, en los que todo se realizaba de forma presencial, utilizando Simulink como generador/monitorizador de las señales necesarias. Sin embargo, hemos detectado que

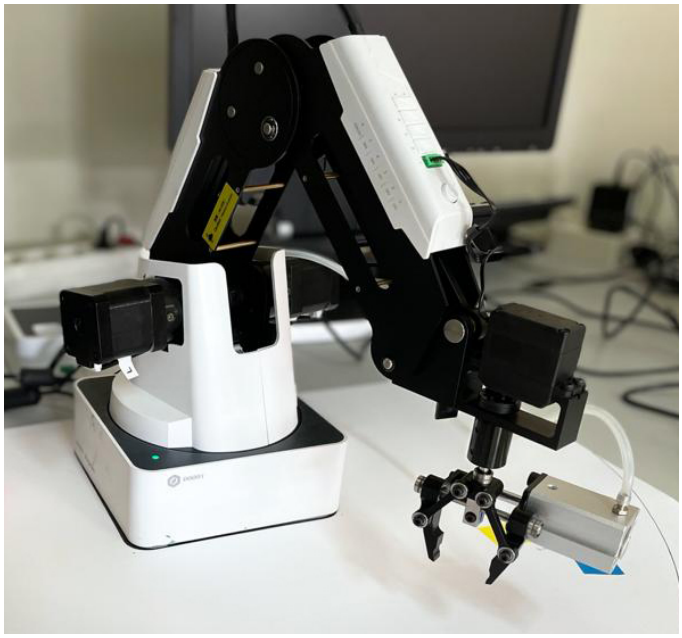


Figura 3.8: Imagen del Dobot Magician

es necesario poner varias plantas del mismo tiempo de forma simultánea, para agilizar el proceso de las pruebas a realizar sobre los controladores continuos, que se ha realizado de forma presencial durante una única sesión de laboratorio.

Finalmente, indicar que información detallada sobre ambas experiencias se recoge en el guión entregado a los alumnos en cada caso, y que se incluye en este documento en los apéndices I y J.

3.2. Laboratorio remoto sobre un brazo robótico

En esta sección describimos un nuevo laboratorio remoto que se ha estado desarrollando a lo largo de este proyecto para las asignaturas de Robótica de diferentes titulaciones de la Facultad de Informática y de la Facultad de Ciencias Físicas.

En este caso, presentamos únicamente la relación entre los diferentes elementos hardware y software que lo constituyen, ya que todavía queda por definir de forma concreta las experiencias que se realizarán sobre el mismo en las diferentes asignaturas.

3.2.1. El sistema real: el robot educativo Dobot Magician

Para realizar las prácticas de control hemos decidido dar acceso remoto a una de los robots manipuladores de los que dispone la Universidad Complutense de Madrid para las asignaturas de Robótica de diferentes titulaciones. En concreto, se ha elegido el brazo robótico Dobot Magician [12], mostrado en la Figura 3.8, ya que es un instrumento educativo cuyos fabricantes permiten que sea controlado a través de un puerto USB 3.0 y una API (Interfaz de Programación de Aplicaciones), accesible desde diferentes lenguajes de programación, que proporcionan para tal efecto.

Más concretamente, el robot está formado por una base, un brazo trasero, un antebrazo y, opcionalmente, un efector final. Además, cuenta con tres o cuatro articulaciones de revolución que le proporcionan un amplio rango de trabajo.

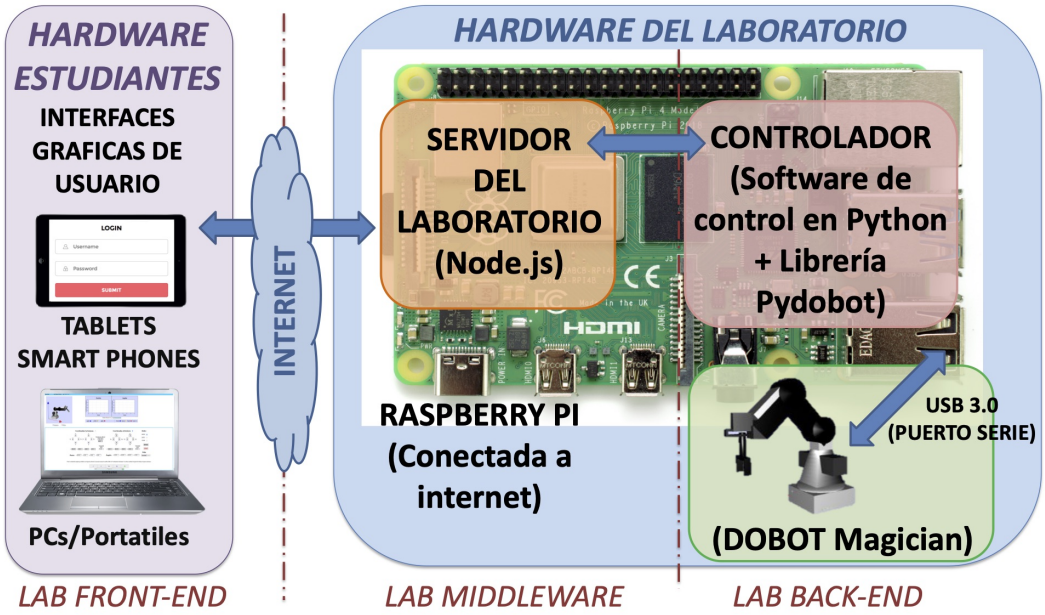


Figura 3.9: Vista general de los elementos involucrados en el laboratorio remoto sobre del Dobot

La API permite moverlo de forma incremental (modo *jogging*), hasta un punto destino (modo Point To Point - *PTP*) o siguiendo un arco (modo *ARC*). Dentro del modo punto a punto, es posible solicitar que el robot siga una linea recta, un movimiento lineal en los ejes articulares, u otro movimiento similar al anterior pero con distancia de aproximación y despegue. También es posible mandar al robot a la posición de inicio (home), hacer una parada de emergencia, y controlar el efector final (una pinza en la imagen de la Figura 3.8).

También es importante indicar que la funcionalidad de esta API es accesible desde diferentes entornos de programación (e.g. C, Python, ROS, Qt, Matlab, Labview, Android, Blockly), motivo por el que sería posible desarrollar un programa controlador en C con una objetivo similar a los programas desarrollados para interactuar con el circuito analógico, es decir, para monitorizar las señales relevantes del dispositivo (en este caso, la posición y orientación del efector final, y los ángulos de las articulaciones) y enviarles las ordenes pertinentes desde la interfaz de la experiencia (en este caso, primitivas de los diferentes movimientos básicos contemplados en la API del Dobot).

Sin embargo, hemos decidido seguir una metodología un poco diferente, sustituyendo el lenguaje de programación del controlador por Python, por dos motivos diferentes. El primero, porque de esta forma el programa de controlador puede utilizar la librería `pydobot`, disponible en [13], para interactuar con el Dobot. La segunda, porque de esta forma estamos empezando a ampliar los lenguajes de programación con los que se puede programar el controlador.

3.2.2. Visión general de las experiencias remotas sobre el Dobot

En la Figura 3.9 se esquematiza la relación que existe entre los elementos hardware y software que se utilizan para hacer una experiencia remota sobre el Dobot Magician. Por una parte, a la izquierda, encontramos el equipamiento de los alumnos, con el que acceden al laboratorio remoto e interactúan con él a través de internet. En la parte central superior situamos la Raspberry PI, sobre la que se despliega el servidor de la página web del laboratorio remoto (que ha sido reconfigurada para este nuevo laboratorio remoto) y el programa controlador (programado en Python).

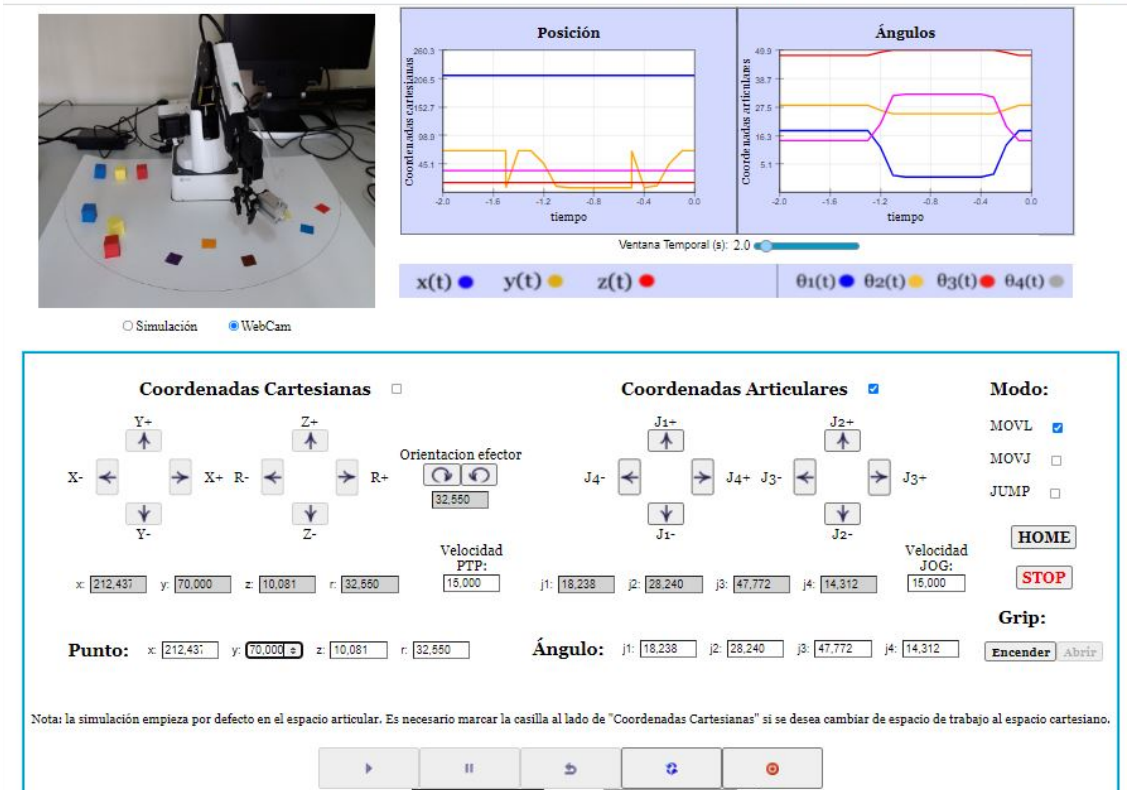


Figura 3.10: Vista general de los elementos involucrados en el laboratorio remoto sobre del Dobot

Finalmente, en la parte inferior derecha de la imagen se esquematiza el intercambio de información entre el programa de control y el Dobot a través del puerto USB de la Raspberry PI y del Dobot.

Como las experiencias con el Dobot requieren de un enfoque diferente a las que se han seguido sobre el circuito analógico, se ha desarrollado completamente, con la herramienta EJsS, una nueva interfaz gráfica de experiencia. En la parte superior izquierda de la interfaz, tal y como se muestra en la Figura 3.10, se puede ver una imagen real del Dobot, que es tomada y enviada a la interfaz de la experiencia en tiempo real, durante el desarrollo de la experiencia, por una cámara que también se encuentra conectada a la Raspberry PI. Además, a su derecha se observa la evolución de las señales significativas del Dobot: la posición y orientación del efector final, y los ángulos de sus cuatro articulaciones. En la parte central se presentan los diferentes elementos que pueden ser utilizados para que los alumnos muevan el robot, utilizando: desplazamientos incrementales en el eje cartesiano o articular, movimientos punto a punto en ambos ejes, o la operación de vuelta al inicio. También se puede parar el robot de forma repentina o modificar el estado del efector final (p.e. abrir o cerrar la pinza). Finalmente, en la parte inferior se utilizan los elementos disponibles en todas nuestras prácticas remotas para ejecutar, parar, resetear o actualizar los parámetros en el código de control de la experiencia.

Por último también conviene indicar que durante el desarrollo de este laboratorio se están poniendo a prueba algunas mejoras adicionales en la metodología de desarrollo del laboratorio. Más concretamente, se están desarrollando nuevos protocolos de comunicación entre la interfaz de la experiencia y el servidor, y entre el servidor y el código de control. La información relacionada con estos nuevos protocolos de comunicación y las experiencias que se pueden desarrollar con este

laboratorio serán documentadas próximamente, una vez que todas los elementos del laboratorio hayan superado las pruebas a las que será sometido por los profesores de las asignaturas de Robótica de las Facultades de Informática y Ciencais Físicas.

3.3. Resumen de los laboratorios actualizados y desarrollados

Tras presentar los laboratorios remotos desarrollados en las dos secciones anteriores, en esta presentamos en la siguiente Tabla un resumen de los mismos, desde el punto de vista de dispositivos de controlan, las asignaturas para las que han sido diseñados, la experiencia concreta que hay que realizar, el proceso seguido para desarrollarlos en este PIMCD, y el curso académico en el que la versión de la experiencia desarrollada en este PIMCD ha sido utilizada por primera vez.

Dispositivo	Asignatura/s	Experiencia	Desarrollo		Curso
			Adaptación	Nuevo	
Circuito	Sistemas Lineales	Respuesta temporal	✓		2019/20
		Diagrama Bode	✓		2020/21
	Procesamiento de Señales	Analisis en Frecuencia Respuesta aleatoria		✓	Próximo
	Control de Sistemas	Red Adelanto	✓		2020/21
		Control PID	✓		2020/21
Robot	Robóticas			✓	Próximo

Además, respecto a las adaptaciones, por una parte conviene destacar que la primera (la respuesta temporal de Sistemas Lineales) ha sido la más sencilla, ya que la práctica se estaba utilizando de forma remota previamente, y solo ha sido necesario adaptar el entorno gráfico de la experiencia correspondiente y configurar los parámetros del servidor desde EJsS, para aprovechar las nuevas posibilidades soportadas por el "Plugin" del laboratorio. Aun así, ha sido el algoritmo sobre el que se han ido probando todas las mejoras de forma sistemática. Por otra, las adaptaciones más complejas han sido las correspondientes a los laboratorios de Control, ya que ha sido necesario implementar en el código del programa de control nuevas versiones de todos los controladores discretos que los alumnos tienen que utilizar en estas experiencias.

Respecto a los desarrollos, conviene indicar también, por una parte, que el laboratorio remoto de Procesamiento de Señales ha sido clasificado como tal, ya que constituye un nuevo material docente que no se utilizaba ni de forma presencial ni remota en los laboratorios con dispositivos reales de la asignatura. Sin embargo, su desarrollo ha sido bastante sencillo, ya que se sólo ha sido necesario incluir nuevas señales de referencia en los laboratorios de Sistemas Lineales. Por otra parte, el laboratorio de robótica ha tenido una complejidad de desarrollo mucho más elevada, ya que se interactúa con un nuevo dispositivo que no se estaba utilizando en los laboratorios remotos hasta este momento, el código del controlador está soportado por un nuevo lenguaje de programación (Python), y los protocolos de comunicación entre los elementos del laboratorio han sido mejorados para este caso.

Finalmente, conviene indicar que estos desarrollos están relacionados con el objetivo **O2.B** de este PIMCD, con el que se perseguía la puesta a punto de laboratorios remotos relacionados con diferentes experiencias. Sin embargo, hay que mencionar que aunque originalmente se deseaba desarrollar experiencias del área de Ingeniería de Sistemas y Automática (ISA) y de Física del Estado Solido, al final solo se han desarrollado nuevas experiencias de ISA. Este recorte en el alcance del objetivo ha sido consecuencia del COVID-19, que ha dificultado el proceso de adaptación del hardware del laboratorio de Físicas de Materiales para hacerlo accesible a la Raspberry PI. Como

contrapartida, los esfuerzos se han focalizado en la mejora general de la metodología de desarrollo de nuevas prácticas.

3.4. Documentación disponible

Las experiencias desarrolladas y presentadas en la sección 3.1 de este capitulo para el circuito analógico se encuentran documentadas, para que los alumnos puedan llevarlas a cabo, por medio de un conjunto de manuales y/o guiones de prácticas, que se presentan en los apéndices de este documento. En la siguiente tabla se indica el apéndice correspondiente a cada experiencia.

Dispositivo	Asignatura/s	Experiencia	Apéndice
Circuito	Sistemas Lineales	Respuesta temporal	F
		Diagrama Bode	G
	Procesamiento de Señales	Analisis en Frecuencia Respuesta aleatoria	H
	Control de Sistemas	Red Adelanto	I
		Control PID	J

Finalmente, recordar que el material didáctico relacionado con el brazo robótico debe ser desarrollado todavía por los docentes involucrados con ese laboratorio, tras analizar que experiencias resultan más adecuadas para la versión actual del laboratorio remoto de robótica.

3.5. Difusión de la metodología y herramientas desarrolladas

El objetivo **O2.C** de este PIMCD contemplaba la difusión de los resultados a través de:

- **Diferentes publicaciones** en revistas y congresos relacionados con el desarrollo de laboratorios remotos en las áreas de Ciencia e Ingeniería.

  A este respecto, queremos indicar que ya se han realizado las dos publicaciones que se mencionan a continuación, según el orden cronológico de publicación:

  1. La comunicación [10] publicada en las XL Jornadas de Automática, dentro de la sesión del grupo de Educación en Control del Comité Español en Automática (CEA). En ella se presentan la idea de los "Plugins" como un nuevo mecanismo de extensión de EJsS y su aplicación para centralizar desde EJsS la gestión de un laboratorio remoto.
  2. El artículo [9] publicado en el número especial "Open Source Software in Learning Environments" de la revista internacional *Electronics*. En él se discuten y detallan múltiples aspectos de los "Plugins" como un mecanismo para mejorar la experiencia de los usuarios de EJsS. Además, se presenta el desarrollo de tres "Plugins" diferentes, para ayudar al lector a entender las posibilidades de la metodología y el proceso seguido para desarrollarlos.

  Además, en la actualidad, se está escribiendo:

  1. Un nuevo artículo, que será próximamente enviado a la revista internacional *IEEE Transactions on Educations*, y que tiene como objeto presentar de forma detallada la versión actual de la metodología que se ha actualizado en este PIMCD para gestionar nuestros laboratorio remotos de bajo coste desde EJsS.

2. Una nueva comunicación, relacionada con el desarrollo del laboratorio remoto sobre un brazo robótico, que será enviada próximamente a las sesiones del grupo de Educación de las próximas Jornadas de Automática.

- **Seminarios en la Facultad de Ciencias Físicas** de la UCM, para publicitar la herramienta entre nuestros colegas más próximos.

  Debido a la situación de la pandemia de la COVID-2019, los desarrollos de este proyecto se han visto ralentizados, y por lo tanto, la disponibilidad de una versión lo suficientemente madura de la herramienta ha sido retardada. Además, la sobre-carga de trabajo asociada a todos los cambios metodológicos en el proceso de enseñanza de la Facultad de Ciencias Físicas que ha ocurrido durante el último año y medio, no hacia recomendable empezar a desarrollar nuevos laboratorios con nuestros colegas en estos momentos.

  Sin embargo, ahora que la herramienta está disponible y que la situación empieza a mejorar, estamos pensando en realizar algún seminario en la Facultad de Físicas con el objeto de poner en marcha nuevas experiencias remotas de laboratorio en nuestro ámbito más cercano.

Además de los métodos de difusión que se consideraban en los objetivos originales del proyecto es importante destacar que:

- También **se ha creado un repositorio [11]** desde el que se puede acceder al "Plugin" de los laboratorios remotos, al código del servidor, al código del software de Control de los sistemas reales, y los proyectos de EJsS desde los que se desarrollan las interfaces gráficas de las experiencias y se gestionan de forma centralizada los laboratorios disponibles.

- **El mecanismo de "Plugins" está disponible desde la versión 6.0 de EJsS.** Esto hace que cualquier usuario que se descargue una versión actual de EJsS pueda utilizar nuestros "Plugins" (o desarrollar otros propios) y por lo tanto, se publicita de forma automática parte de los avances de este proyecto entre la comunidad educativa que utiliza habitualmente esta herramienta para desarrollar simulaciones interactivas con EJsS.

Capítulo 4

Conclusiones

Este Proyecto de Innovación Educativa y Mejora de la Calidad de Docente tenía como objetivo general automatizar el proceso de generación y puesta en marcha de los laboratorios remotos de bajo coste que el área de Ingeniería de Sistemas y Automática viene desarrollando desde hace varios años, y que se encuentran actualmente soportados por una Raspberry PI como elemento de despliegue del software del laboratorio remoto, Node.js como entorno de desarrollo del servidor de las páginas web del laboratorio, y EJsS como herramienta de desarrollo de las interfases gráficas de las experiencias, mediante la que los alumnos interactúan desde sus equipos y a través de Internet con el hardware del laboratorio.

En este capítulo final analizaremos el grado de cumplimiento de cada uno de los objetivos planteados al inicio del proyecto y propondremos algunas lineas futuras de actuación.

4.1. Cumplimiento de objetivos

El objetivo principal de este proyecto es el desarrollo de una herramienta integral y de bajo coste para dar acceso remoto, a través de Internet, a las experiencias de los laboratorios de las áreas de Ciencias e Ingeniería.

Como herramienta integral indicamos que ésta debía permitir realizar todas las labores necesarias para poner en marcha las experiencias del laboratorio desde un mismo entorno, cosa que se ha logrado ya que EJsS se ha convertido (gracias al nuevo mecanismo de "Plugins" desarrollado en este proyecto e incorporado en EJsS a partir de su versión 6.0, y al Plugin implementado para nuestros laboratorios remotos) en la herramienta de desarrollo y gestión de todo el laboratorio remoto.

El requisito de bajo coste de conexión de una experiencia a Internet, salvo para casos especiales de sensorización y actuación especiales, se logra ya que el software (EJsS, Node.js, y controladores programados en lenguajes estándar) que da soporte a la conexión es gratuito y está desplegado sobre una Raspberry PI (que tiene un coste inferior a 110€, cuando se incluye en este valor el de la memoria necesaria para desplegar su software).

De forma más concreta, el objetivo anterior se articula en los dos grupos de objetivos que se detallan y analizan en las dos secciones siguientes.

4.1.1. Centralización del proceso de generación de los laboratorios remotos

Este grupo recoge los objetivos encargados de automatizar e integrar en una única herramienta el proceso de generación de los laboratorios remotos de bajo coste de la UCM basados en EJsS, Node.js y la Raspberry PI. Para lograrlo, el proyecto planteaba:

O1.A) *Facilitar la conexión entre las variables utilizadas en el código del controlador (que interactúa con los sensores y actuadores conectados al dispositivos) y las variables javascript de la página web que da acceso al laboratorio.*

Este objetivo se ha logrado, ya que el "Plugin" del laboratorio remoto permite, por una parte, que desde EJsS se configuren las variables de intercambio de información a través del servidor y, por otra, incluye un nuevo Elemento de Modelo que puede ser utilizado por la interfaz gráfica de la experiencia para acceder a la información de estas variables de forma automática.

O1.B) *Permitir acceder y editar el código del controlador de la Raspberry PI desde EJsS, y una vez que esté listo, descargarlo y compilarlo desde EJsS sobre la Raspberry PI.*

Este objetivo también se ha logrado, ya que el "Plugin" del laboratorio remoto permite que desde EJsS se realicen todas las labores anteriormente indicadas. Aún más, no sólo permite descargar y modificar el código del controlador en EJsS, si no que también permite que se descargue el makefile utilizado para compilarlo, de forma que el desarrollador también pueda controlar y modificar el proceso de creación de la aplicación que interactua directamente con los dispositivos del laboratorio.

O1.C) *Posibilitar el volcado de la página web de HTML5.0 y javascript creada en EJsS por el responsable del laboratorio para que el estudiante pueda interactuar desde su PC, móvil o tableta con la experiencia que debe realizar.*

Este objetivo también se ha logrado, ya que el Plugin del laboratorio permite descargar el entorno gráfico de la experiencia desarrollado en EJsS sobre la Raspberry PI, y hacer que este sea directamente reutilizable por el servidor en la próxima conexión del alumno.

Además, cabe destacar que este objetivo ha sido ampliado con una nueva posibilidad, mediante la que el desarrollador puede también probar el entorno gráfico de la experiencia desde su equipo, antes de desplegarlo en el laboratorio remoto. Esto agiliza el desarrollo de nuevos labotorios remotos, ya que solo se desplegarán sus interfaces después de que su desarrollador compruebe que se comportan de forma adecuada sobre la experiencia.

O1.D) *Configurar y descargar la página web de instrucciones del laboratorio remoto.*

Este objetivo también se ha logrado, ya que a la vez que se despliega la interfaz gráfica de la herramienta, se descarga la página de Descripción de la misma que ha sido desarrollada en EJsS.

O1.E) *Configurar la información que debe ser almacenada en los ficheros de datos de cada experimento.*

Este objetivo también se ha conseguido, ya que en el fichero de configuración de las variables que se intercambian con el servidor, al que se tiene acceso desde EJsS gracias al nuevo "Plugin" desarrollado, también se indican cuales deben ser almacenadas en el fichero de datos de cada experiencia.

O1.F) *Simplificar el proceso de alta y baja de usuarios (alumnos) a la página web del laboratorio.*

Este objetivo también se ha logrado, ya que el "Plugin" desarrollado para el laboratorio permite que desde EJsS se pueda dar de alta/baja a diferentes alumnos. De formas más concreta, se puede obtener el listado de alumnos disponible en el servidor, modificarlo, cargar un fichero en formato CSV, y descargar el nuevo listado de alumnos y sus permisos en el servidor del laboratorio.

Por lo tanto, se han cumplido todos los objetivos del primer grupo, de forma que la nueva metodología permite centralizar la gestión del laboratorio desde EJsS. Aun más, conviene resaltar de nuevo, que no se ha hecho una modificación ad-hoc de la funcionalidad y el entorno gráfico de EJsS, sino que se ha ido un paso más allá: se ha dotado a EJsS con una nueva funcionalidad que permite a cualquier usuario con conocimientos de programación e interés en adaptar EJsS a sus necesidades, realizar sus modificaciones y encapsularlas en un "Plugin" propio, que puede ser distribuido fácilmente y reutilizable en nuevas versiones de EJsS. Esta modificación de EJsS, inicialmente no contemplada en este proyecto, ha sido incorporada a partir de la versión oficial 6.0 de dicha herramienta, y está sirviendo para articular las nuevas versiones de dicha herramienta en conjuntos de funcionalidades diferentes para distintos tipos de usuarios.

4.1.2. Extensión del uso de la herramienta a experiencias de diferentes asignaturas

Este grupo de objetivos se centra en lograr que el uso de la herramienta se extienda a las experiencias de laboratorio de diferentes asignaturas de Ciencias e Ingeniería. Para lograrlo, el proyecto planteaba llevar a cabo las siguientes labores:

O2.A) *Documentar el funcionamiento de la nueva herramienta, para que los responsables de los laboratorios dispongan de la información necesaria para crear desde cero un laboratorio remoto.*

Para dar respuesta a este objetivo se han desarrollado dos manuales, uno en el que se indica el proceso de instalación del software necesario para poder desarrollar los laboratorios remotos y otro en el que se presenta el proceso seguido para poner a punto un laboratorio desde cero. De forma complementaria, también se han desarrollado unos videos en los que se detallan las partes más relevantes del proceso.

Además, también se han documentado la mayoría de las experiencias que se han actualizado/desarrollado a lo largo del PIMCD, para que sirvan de ejemplo a posibles usuarios de la herramienta.

O2.B) *Pre-seleccionar y poner a punto los laboratorios remotos de un conjunto experiencias del área de Ingeniería de Sistemas y Automática (ISA) y de Física del Estado Sólido.*

Este objetivo se ha conseguido de forma parcial. Se ha puesto apunto (actualizando o desarrollando desde su comienzo) 5 experiencias de ISA, relacionadas con las asignaturas de Sistemas Lineales, Procesamiento de Señales y Control de Sistemas, y se ha desarrollado el software necesario para que los alumnos también puedan interactuar con un brazo robótico a través de Internet desde sus equipos.

Queda pendiente desarrollar las experiencias concretas que se realizarán sobre el brazo robótico y otras relacionadas con el área de Física del Estado Solido. Estas últimas han sido pospuestas, ya que la conexión de los dispositivos de las experiencias que se deseaba poner en remoto con la Raspberry PI se ha visto dificultada por la restricciones de la pandemia.

O2.C) *Publicitar y publicar la nueva herramienta que se desarrolle en los foros adecuados.*

Para lograr este objetivo se han realizado dos publicaciones:

1. La comunicación [10] publicada en las XL Jornadas de Automática, dentro de la sesión del grupo de Educación en Control del Comité Español en Automática (CEA). En ella se presentan la idea de los "Plugins" como un nuevo mecanismo de extensión de EJsS y su aplicación para centralizar desde EJsS la gestión de un laboratorio remoto.

2. El artículo [9] publicado en el número especial "Open Source Software in Learning Environments" de la revista internacional *Electronics*. En él se discuten y detallan múltiples aspectos de los "Plugins" como un mecanismo para mejorar la experiencia de los usuarios de EJsS. Además, se presenta el desarrollo de tres "Plugins" diferentes, para ayudar al lector a entender las posibilidades de la metodología y el proceso seguido para desarrollarlos.

Además, en la actualidad se están preparando dos publicaciones adicionales:

1. Un nuevo artículo, que será próximamente enviado a la revista internacional *IEEE Transactions on Educations*, y que tiene como objeto presentar de forma detallada la versión actual de la metodología que se ha actualizado en este PIMCD para gestionar nuestros laboratorio remotos de bajo coste desde EJsS.
2. Una nueva comunicación, relacionada con los desarrollos del laboratorio remoto sobre el brazo robótico, que será enviada próximamente a las sesiones del grupo de Educación de las próximas Jornadas de Automática.

Por otra parte, es importante destacar que también se ha creado un repositorio [11] desde el que se puede acceder al "Plugin" de los laboratorios remotos, al código del servidor, al código del software de control de los sistemas reales, y los proyectos de EJsS desde los que se desarrollan las interfaces gráficas de las experiencias y se gestionan de forma centralizada los laboratorios disponibles.

Finalmente, cabe indicar que ha quedado pendiente la difusión de la existencia de la herramienta dentro del ámbito más próximo, el de la propia Facultad de Ciencias Físicas, para explorar la posibilidad de desarrollar nuevas experiencias con esta metodología en un futuro próximo.

Por lo tanto, podemos concluir que los objetivos de este grupo no han sido cumplidos plenamente, principalmente debido a las dificultades que la pandemia ha añadido a nuestras labores docentes y de investigación habituales. Aun así, podriamos considerar que en conjunto se ha logrado el espiritu de los mismos, ya que se está consiguiendo publicitar la herramienta a una amplia comunidad a través de las publicaciones y se ha extendido su uso a dos nuevas asignaturas (Procesamiento de Señales y Robótica) en la que no estaban siendo utilizada previamente.

4.2. Trabajo futuro

Aunque se haya cumplido con la mayor parte de los objetivos planteados en el proyecto, la metodología y herramienta admite diferentes mejoras que serán tenidas en cuenta en sus próximas versiones.

Por una parte, estas mejoras son ergonómicas y vienen solicitadas por sus usuarios actuales (diferentes profesores y algunos alumnos que la han estado utilizando en sus trabajos fin de grado). A modo de ejemplo, podría ser útil que el usuario no sólo pueda descargarse el código en C desde EJsS para modificarlo, si no también crear nuevos ficheros para articular mejor su código, borrar versiones antiguas, etc. También sería conveniente que en la zona de mensajes de EJsS se muestren más mensajes que permitan al usuario saber que está sucediendo mientras realiza diferentes labores en las que se interactúa con el servidor.

Por otra parte, y tal y como ya se ha mencionado, es necesario desarrollar algunas prácticas de Física del Estado Solido y publicitar la herramienta entre nuestros colegas más cercanos, ya que la puesta a punto de laboratorios remotos es una estrategia educativa que permite a los alumnos

acceder al material durante más horas y continuar con sus experiencias después de los laboratorio presenciales. De hecho, su existencia ha facilitado el desarrollo de nuestra docencia durante las condiciones excepcionales en las que nos hemos encontrado debido a la pandemia de la COVID-19.

Finalmente, también queremos analizar que estrategias resultan más adecuadas para unificar, de una forma amigable y transparente para los alumnos, sus experiencias en los laboratorios remotos y presenciales, de forma que puedan llegar al laboratorio presencial con una parte de la experiencia ya preparada y continúen con experiencias complementarias, que les permitan reforzar sus conocimientos desde cualquier equipo conectado a Internet.

Apéndice A

Instalación del software necesario para el Laboratorio Remoto

El software necesario para hacer funcionar el laboratorio remoto se encuentra disponible en el repositorio [11]. En esta guía se describe la instalación del servidor *ReNoLabs* sobre una *Raspberry PI*, ya que es la forma más habitual de desplegar el servidor. Sin embargo, el procedimiento debería ser igualmente válido para cualquier otro sistema basado en GNU/Linux. A lo largo del documento se asume que tenemos acceso directo o remoto a la Raspberry con permisos de superusuario. En casos de no ser así, es preciso seguir la guía de instalación que se referencia en el apartado de requisitos.

A.1. Requisitos

En esta sección se describen los requisitos que el software y hardware del laboratorio deben cumplir para poder poder desplegarlo con éxito sobre una Raspberry PI.

A.1.1. Requisitos software

Las versiones mínimas del software recomendadas, con las que se ha verificado el correcto funcionamiento del servidor, son las siguientes.

- Raspbian GNU/Linux 10.
- Node.js v10.21.0 (o superior).
- npm 5.8.0 (o superior).
- git 2.20 (o superior).
- OpenSSH 7.9 (o superior).

Para instalar el sistema operativo Raspbian GNU/Linux se recomienda consultar la siguiente página web [14] y utilizar la imágen *con escritorio y software recomendado*, ya que tiene pre-instalado el software mencionado en requisitos.

Para para poder acceder a la Raspberry PI de forma remota a través de una conexión SSH se utiliza el software OpenSSH, que se distribuye pre-instalado en todas las imágenes de Raspbian. No obstante, por defecto se encuentra desactivado, siendo necesario activar la conexión explícitamente. Esto se puede hacer a través de la interfaz gráfica, durante el primer arranque después de la

instalación del sistema operativo. También es posible hacerlo de forma manual, añadiendo un archivo vacío con nombre "ssh.[en] el sistema de archivos etiquetado como `boot` en la tarjeta SD. Para más información, el recomentable consultar la siguiente pagina web [15]

En caso de que se utilice una imágen del sistema operativo Raspbian GNU/Linu que no incluya las aplicaciones requeridas, éstas pueden instalarse de forma manual mediante la instrucción:

```
sudo apt install node.js npm git
```

A.1.2. Requisitos hardware

Existen varias versiones diferentes de Raspberry PI. Para nuestro laboratorio es aconsejable utilizar al menos la versión *Raspberry PI 2 B+*.

Esto se debe a que, aunque es posible hacer funcionar el servidor Raspberry PI 1, no es recomendable su uso, ya que puede presentar problemas de compatibilidad con Node.js v10, y a la menor capacidad de procesamiento frente a modelos más recientes. Por otra parte el modelo Raspberry PI 2B+ en general proporciona un rendimiento adecuado para las necesidades del laboratorio remoto. No obstante, en caso de que se necesite más potencia de cálculo o por razones de conectividad (p.e. en caso de que se requiera conexión Bluetooth o WiFi) puede ser más conveniente utilizar una Raspberry PI 3 o 4.

A.2. ReNoLabs

En esta sección indicamos como instalar el servidor de ReNoLabs en la Raspberry PI y como adaptar el comportamiento de EJsS para que pueda interactuar con él.

A.2.1. Instalación y puesta a punto del servidor

El primer paso consiste en descargar el código del servidor en la Raspberry PI. Para ejecutar el servidor es necesario disponer de una instalación reciente de *Node.js* y opcionalmente, *npm* (Node.js Package Manager), que nos ayudará a instalar algunas dependencias, y *git*, para descargar el código del repositorio [11] o alternativamente, si se desea obtener la versión más actual del servidor del laboratorio remoto de [16]. Tal y como hemos mencionado en la sección anterior, las versiones más recientes del sistema operativo Raspbian OS ya incluyen el software mencionado.

Para descargar y empezar a ejecutar el servidor abrimos un emulador de terminal y ejecutamos las siguientes órdenes:

```
raspberry:~/$ git clone https://github.com/jcsombria/ReNoLabs
Clonando en 'ReNoLabs'...
remote: Enumerating objects: 448, done.
remote: Counting objects: 100% (194/194), done.
remote: Compressing objects: 100% (120/120), done.
remote: Total 448 (delta 79), reused 159 (delta 59), pack-reused 254
Recibiendo objetos: 100% (448/448), 3.06 MiB | 10.74 MiB/s, listo.
Resolviendo deltas: 100% (155/155), listo.
```

A continuación, instalamos las dependencias declaradas por el servidor:

```
raspberry:~/$ cd ReNoLabs
raspberry:~/$ npm install
...
added 196 packages from 117 contributors in 14.842s
```

Y finalmente, ejecutamos el servidor:

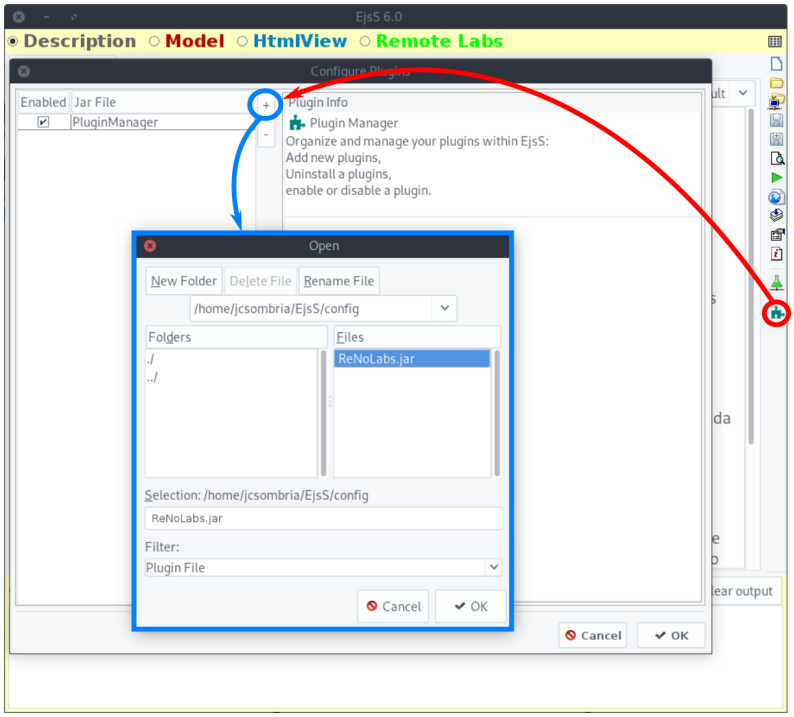


Figura A.1: Instalación del "Plugin" de ReNoLabs en EJsS.

```
raspberry:~/$ nodejs server.js
2021-06-25T14:02:49.226Z info:  Server started on http://127.0.0.1:80
```

A partir de este momento, ReNoLabs está en línea, y por defecto queda a la escucha en el puerto 80 (HTTP) del *localhost*. Además, incluye una base de datos de usuarios mínima, en la que únicamente aparece el usuario `admin` con password `admin`, para que se pueda acceder al servidor desde la herramienta EJsS mejorada. Finalmente, también es capaz de ejecutar de forma predeterminada el módulo controlador local escrito en C.

A pesar de que esta configuración básica deberá ser modificada para adaptarla a las necesidades de cada uno de nuestros laboratorios, en este punto se cuenta con un sistema funcional, que proporciona todo la infraestructura software necesaria para dar soporte a la creación y despliegue de un laboratorio remoto.

A partir de este punto, toda la configuración del laboratorio (p.e. la definición del controlador y la modificación de la base de datos de usuarios) se puede realizar de forma remota a través del "Plugin" ReNoLabs for EJsS.

A.2.2. Instalación y puesta a punto de EJsS

En este caso, el primer paso consiste en descargarse la versión más actualizada de EJsS desde la página web de sus desarrolladores [17]. A continuación hay que instalar el "Plugin" desarrollado para gestionar el laboratorio remoto desde EJsS, que está disponible en el repositorio de este PIMCD [11].

El "Plugin" se distribuye en forma de un único archivo java empaquetado, con nombre *ReNoLabs.jar*. La instalación se puede hacer a través del instalador de "Plugins" incorporado en EJsS siguiendo el proceso descrito en la sección 2.1.7. Para ello, tal y como se representa en la Figura A.1,

se debe hacer click en el botón del instalador de “Plugins” (resaltado en rojo), y a continuación presionar el botón de adición del “Plugin” (resaltado en azul), seleccionar el archivo del “Plugin” desarrollado en este PIMCD, y pulsar aceptar. Finalmente, EJsS solicitará al usuario reiniciar la aplicación para recargar el “Plugin” y poder comenzar a utilizarlo.

También es posible realizar la instalación del “Plugin” de forma manual. Para ello basta con copiar el archivo *ReNoLabs.jar* a la carpeta `$EJS_WORKSPACE/config/CustomPlugins`, siendo $EJS_WORKSPACE la ruta al espacio de trabajo configurado por cada usuario en su instalación de EJsS.

Apéndice B

Manual de usuario del EJsS adaptado para los laboratorios remotos

En este apéndice se explica el proceso para poner en marcha y desplegar un nuevo laboratorio remoto desde cero. Para poder hacerlo, hay que tener disponer de una Raspberry PI adecuada, equipada con el software que se especifica en el apéndice anterior, y un PC con una versión de EJsS igual o superior a las 6.0 y con el "Plugin" de RenoLabs instalado.

Los pasos que se deben realizar se presentan en un orden que consideramos adecuado para la exposición, aunque en realidad no existe la necesidad de ejecutarlos secuencialmente. Mas bien se trata de un proceso iterativo, en el que los diferentes aspectos de la configuración del laboratorio van evolucionando en paralelo.

B.1. Añadir y acceder a una nueva instancia de laboratorio

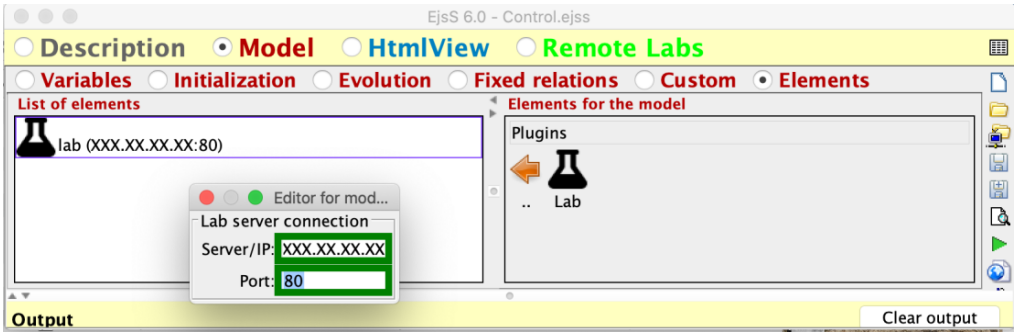
El primer paso para configurar una conexión al servidor del laboratorio remoto es añadir una nueva instancia del elemento `Lab` proporcionado por la extensión. Esta instancia representa nuestro laboratorio y proporciona métodos que nos permiten comunicarnos con él desde nuestra aplicación en EJsS.

A continuación, debemos configurar el acceso (*URL*) al servidor, introduciendo el valor adecuado en los campos resaltados en verde en la Figura B.1a.

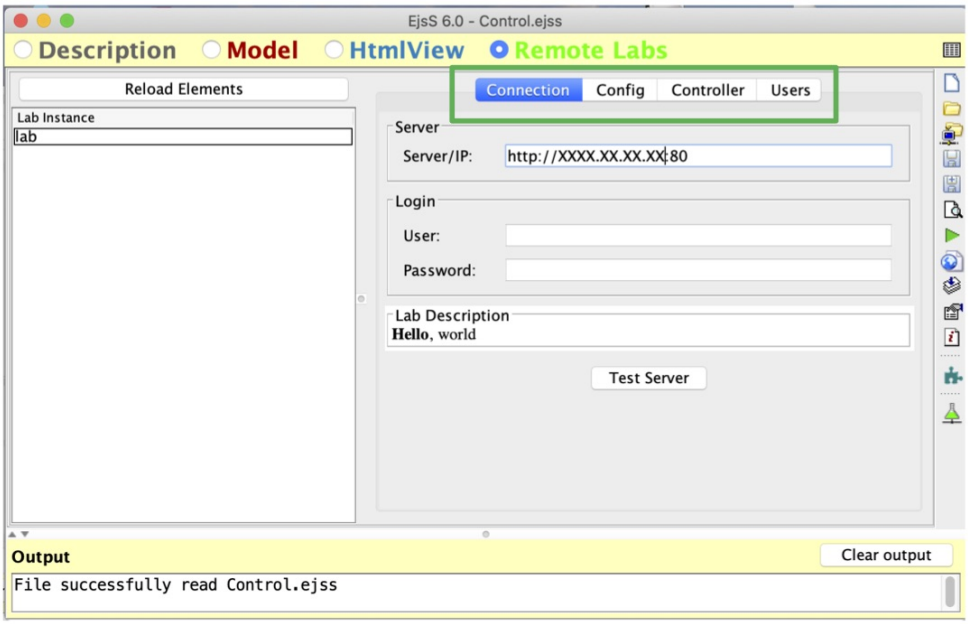
Finalmente, debemos ir a la pestaña de Laboratorios Remotos del panel Principal, tal y como se muestra en la Figura B.1b, cargar los elementos del laboratorio (pulsando el botón Reload Elements), pinchar sobre dicho laboratorio, y a continuación introducir las credenciales de usuario – que necesita permisos de administrador para poder ver/editar la configuración del servidor de forma remota –. Finalmente, se debe presionar el botón *Test Server* para verificar que el servidor se encuentra accesible.

B.2. Ajuste de la configuración

El siguiente paso es acceder a los archivos de configuración a través de la pestaña `config` (ver Figura B.2). Para ello, hay que presionar el botón *Get Config* de forma que a continuación se obtienen los archivos de configuración del laboratorio. Estos se mostrarán en el editor de configuración, y consisten en una estructura JSON que contiene todas las opciones de configuración como la URL del servidor, módulos hardware, opciones de compatibilidad, etc. Esta característica, aunque enormemente útil, debe ser utilizada con precaución: en caso de que la configuración se haga de forma incorrecta es posible provocar que el servidor no funcione correctamente, e incluso existe la



(a) Configuración de la instancia del elemento *Lab* para acceder al laboratorio remoto.



(b) Introducción del usuario para gestionar el laboratorio remoto

Figura B.1: Configuración inicial del laboratorio remoto

posibilidad de perder acceso remoto al servidor. En este último caso, sería necesario conectarse a la Raspberry PI por SSH, y relanzar de forma manual el servidor.

La configuración del laboratorio remoto se distribuye en tres archivos diferentes:

- *AppConfig.js*, contiene las opciones globales de configuración, como son la IP en la que escucha del servidor, la duración de cada sesión (durante la que el alumno interactúa de forma continuada con el dispositivo del laboratorio), o la habilitación de soporte a RIP (que es un protocolo de comunicación que permite conectar el servidor a otros LMS – Learning Management Systems – más complejos, como Moodle).

- *LabConfig.js*, define la práctica, por medio de la configuración de información sobre señales/parámetros, código del controlador, variables que se deben almacenar durante la experiencia, etc. Está configuración se proporciona en un archivo diferente con el propósito de poder cambiar fácilmente de una práctica a otra sin afectar a otros aspectos de la configuración.

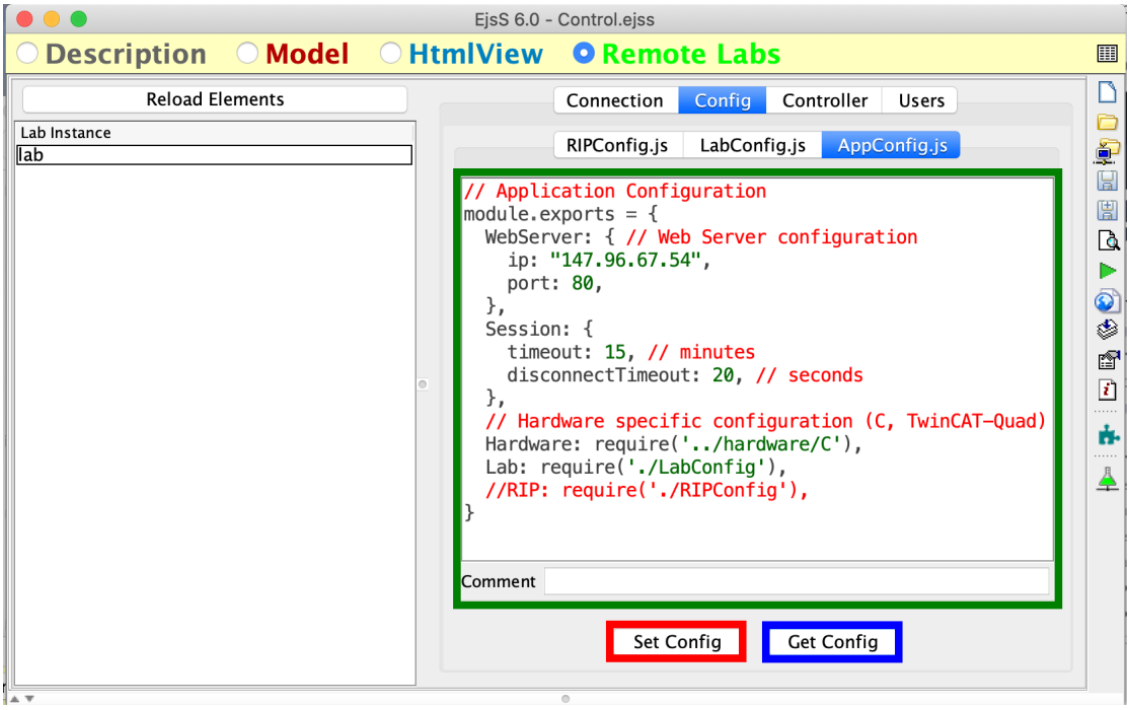


Figura B.2: Edición de la configuración del laboratorio remoto desde EJsS.

- *RIPConfig.js*. Este archivo contiene configuración específica de RIP, para proporcionar soporte para la integración de nuestro laboratorio con otras plataformas educativas que soporten dicho protocolo de comunicación.

La configuración global por defecto se muestra a continuación:

```
// Application Configuration
module.exports = {
  WebServer: { // Web Server configuration
    ip: "127.0.0.1",
    port: 80,
  },
  Session: {
    timeout: 15, // minutes
    disconnectTimeout: 20, // seconds
  },
  // Hardware specific configuration (C/WiringPi, TwinCAT-ADS, Python/Dobot)
  Hardware: require('../hardware/C'),
  Lab: require('./LabConfig'),
  //RIP: require('./RIPConfig'),
}
```

En la sección *WebServer* se indica al servidor que debe quedar a la escucha en el puerto 80 (HTTP) en la dirección 127.0.0.1. De esta manera, el servidor solo aceptará conexiones locales, lo que puede ser útil en caso de que se desee realizar una conexión encriptada a través de un tunel SSH. En caso

de que queramos dar acceso a través de una IP o puerto diferente, deberemos modificar los campos *ip* o *port*. En la Figura B.2 se muestra la edición de esta configuración desde EJsS.

La siguiente sección, *Session*, indica el tiempo máximo de duración de la sesión (*timeout*) y el tiempo de espera para reconexión (*disconnectTimeout*), tras el cuál se considera que el usuario se ha desconectado definitivamente y se concluye la sesión.

Finalmente, los campos restantes son la configuración del módulo hardware utilizado (*Hardware*), la configuración de la práctica (*Lab*) y la configuración de RIP. En este caso, el soporte RIP está deshabilitado por defecto.

Respecto a la configuración de la práctica en el archivo 'LabConfig.js', se muestra un ejemplo a continuación:

```
// Lab configuration
module.exports = {
  GUI_JS: 'circuitos_CS.js',
  controller: 'circuitos',
  info: {
    name: 'Circuitos - Configuración PID',
    description: '<h1>Práctica de Circuitos</h1><p>Lab description</p>',
  },
  parameters: {
  ... // No mostrado aquí
  },
}
```

Los campos más relevantes son:

- *GUI_JS*: referencia a la aplicación del laboratorio remoto generada con EJS.
- *controller*: referencia al archivo ejecutable principal del controlador que será ejecutado durante la experiencia remota y cuya preparación se describe en la la sección B.2.1.
- *info*: título informativo de la experiencia que se mostrará en la página de acceso a ReNoLabs.
- *parameters*: descripción de los parámetros para la generación automática de la interfaz (cuya información se detalla en la sección B.2.3 de este documento).

B.2.1. El controlador

El servidor de laboratorio da soporte a la definición de un controlador configurable. Actualmente, se proporcionan tres tipos de controlador hardware: C/WiringPi, TwinCAT ADS y Python/Dobot. En esta guía nos centramos en explicar el funcionamiento del controlador `C/WiringPi`. En cualquier caso, el procedimiento de configuración desde EJsS será similar para cualquier otro lenguaje/plataforma, aunque las funciones soportadas pueden variar ligeramente (por ejemplo, el controlador TwinCAT ADS no soporta la programación remota que se describe a continuación).

La pestaña *controller* (mostrada en la Figura B.3) proporciona un editor de código para realizar tareas de mantenimiento del controlador desde EJsS. Mediante el botón *Get Code* se solicita al servidor el envío de los archivos del controlador (resaltado en azul), que se cargan y se muestran en el editor (en verde). Este editor permite ver el código y realizar las modificaciones que el responsable del laboratorio considere adecuado. Una vez hechos los cambios, para notificar la actualización al servidor es preciso presionar el botón *Set Code* (resaltado en rojo). Al enviar los cambios, el controlador se compila en el servidor, y en caso de encontrar algún error se notificará al usuario

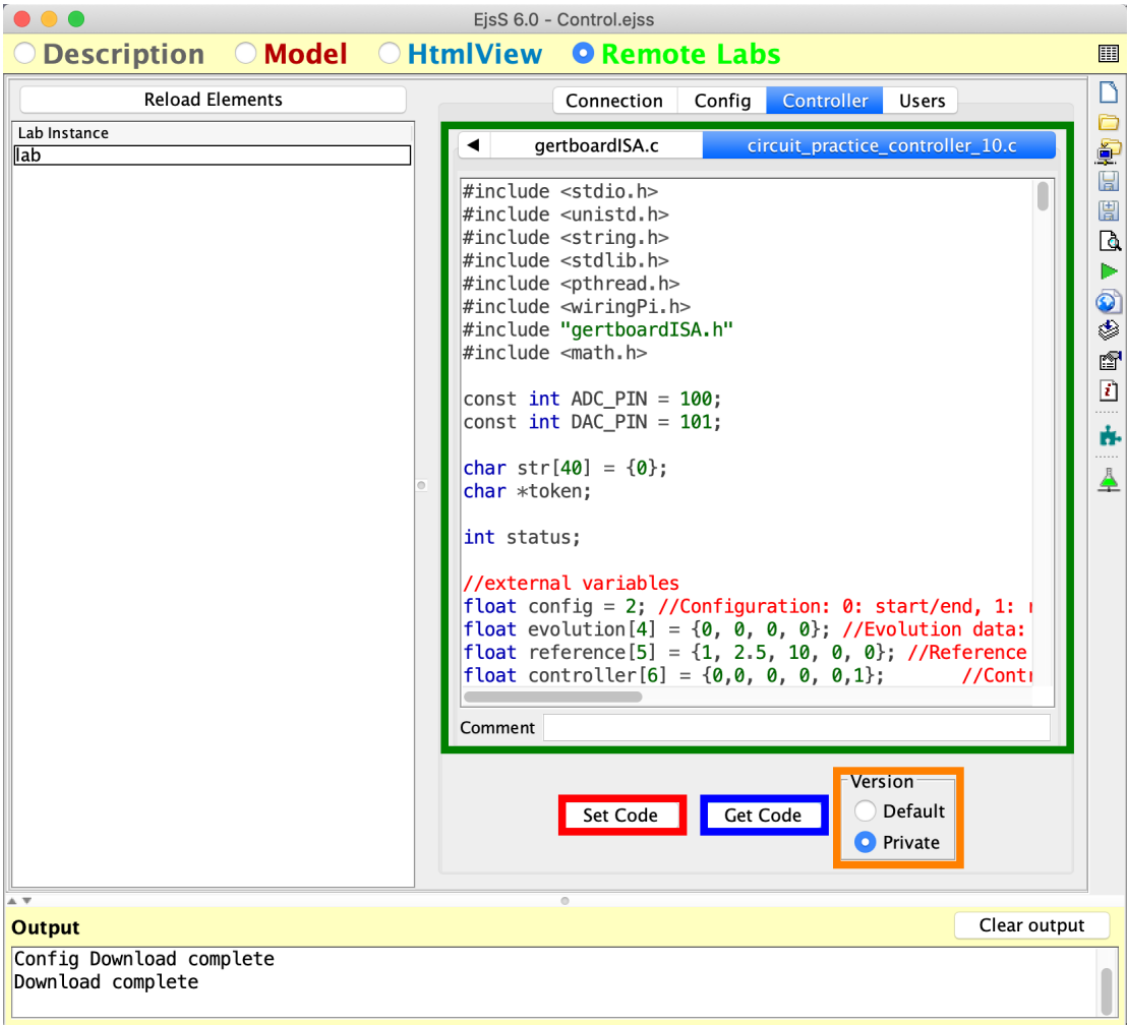


Figura B.3: Actualización del controlador del laboratorio remoto desde EJsS.

a través de la interfaz de mensajes de EJsS. Por último, es posible seleccionar entre la versión *por defecto* y la versión *privada* (resaltadas ambas opciones en naranja sobre la Figura B.3). El servidor almacena una versión *por defecto*, que se utiliza como controlador de referencia y solo puede ser modificada por un usuario administrador. Opcionalmente, y dependiendo del laboratorio, cada usuario puede disponer de una copia *privada* del controlador y modificarla a su antojo. Por una parte, el controlador por defecto solo se modifica, generalmente, durante la fase de desarrollo o en tareas de mantenimiento. Por la otra, la versión privada sirve para proporcionar una versión estable de referencia o una estructura (a modo de plantilla) que el alumno debe modificar como parte de la experiencia. Esta segunda opción no ha sido utilizada todavía con los alumnos, ya que solo debería estar disponible para alumnos de cursos avanzados, ya que el desarrollo de un código erróneo impide que se pueda ejecutar el controlador del alumno, y la introducción de código malicioso puede desconfigurar totalmente el servidor de prácticas.

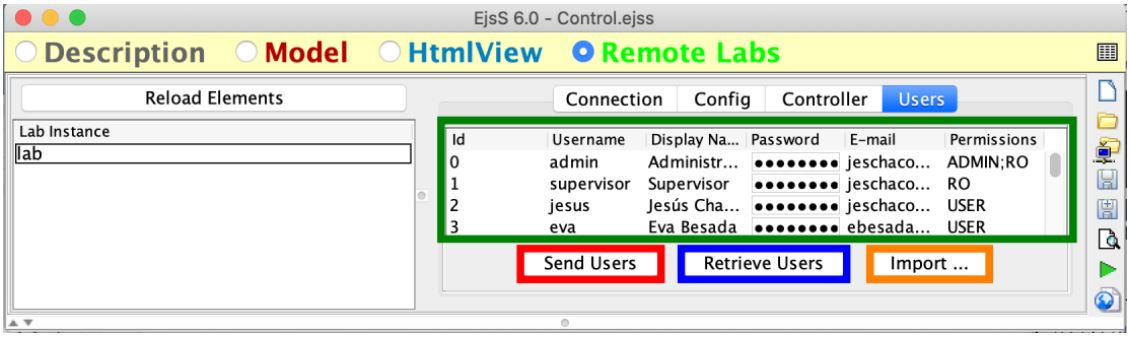


Figura B.4: Edición de los usuarios del laboratorio remoto en EJsS.

B.2.2. Los usuarios

Es posible dar de alta o modificar usuarios del laboratorio desde EJsS. Para ello hay que seleccionar la pestaña del editor de usuarios, tal y como se muestra en la Figura B.4. El editor de usuarios muestra una tabla con los diferentes campos del perfil de usuario (*id*, *username*, *password*, etc.). Mediante el botón *Retrieve Users* (resaltado en azul) se solicita al servidor la lista de usuarios registrados. A continuación es posible editar directamente los usuarios en la tabla, en caso de que se desee realizar cualquier modificación. Finalmente, para notificar los cambios al servidor es preciso presionar el botón *Send Users* (resaltado en rojo).

Otra opción complementaria es importar los usuarios desde un archivo CSV. Para ello, será necesario pulsar el botón *Import...* (resaltado en naranja) y seleccionar el archivo con la información de los usuarios. A continuación, opcionalmente, se puede especificar qué símbolo se desea utilizar como separador (por defecto, ','). Una vez importada/modificada la información de los usuarios se deberá realizar el envío de los datos modificados, al igual que antes.

El servidor recarga automáticamente la base de datos de usuarios tras la actualización, por lo que hay que tener en cuenta que en caso de modificar de forma inadecuada el usuario administrador (por ejemplo, retirando los permisos de administración), existe la posibilidad de perder acceso al laboratorio remoto.

B.2.3. La interfaz gráfica del laboratorio remoto

Hasta ahora toda la configuración realizada guarda relación con la infraestructura requerida para garantizar que el laboratorio se encuentra en funcionamiento, y es posible comunicarnos y sincronizar datos con él. Por último, aunque no menos importante, es necesario construir una interfaz gráfica para los estudiantes, es decir, proporcionar una aplicación que muestre información sobre la plataforma experimental y les permita realizar la experiencia correspondiente a la práctica, y que les permita acceder a los datos que se han generado durante la experiencia. Afortunadamente, la mayor parte del trabajo de creación de la interfaz de usuario está automatizado, de manera que ReNoLabs proporciona, a partir de la configuración en el servidor, un conjunto de controles e indicadores que permiten modificar las variables del laboratorio y controlar la ejecución de las experiencias.

En la mayoría de los laboratorios remotos, la distribución de los elementos de la interfaz es similar, y no presenta mucha complejidad: una representación del sistema (ya sea a través de una cámara web, realidad aumentada o la representación que se desee mostrar), algunas gráficas que muestran la evolución de los estados y variables de interés del sistema, y por último algunos controles que permitan interactuar con el sistema. En este sentido, el plugin ReNoLabs para EJsS

proporciona tres elementos de la vista (*View Elements*) que pueden resultar útiles:

- *LabLogin* (), añade a la interfaz un conjunto de campos usuario/contraseña y un botón de conexión. Este elemento es útil para poder realizar el acceso a la interfaz del laboratorio mientras está siendo desarrollado, y no es necesario cuando la aplicación se aloja en ReNoLabs y está siendo utilizada por los estudiantes.

- *LabControl* (), implementa el modelo de ejecución del laboratorio ReNoLabs, proporcionando diferentes botones para controlar el estado de ejecución (*inicio*, *pausa*, *reinicio*, *actualización*, *conexión/desconexión*).

- *LabFunctionParameter* () proporciona la construcción automática de los controles/indicadores que permiten acceder a las variables del laboratorio. Por este motivo es es quizás el componente de mayor interés. En concreto, este elemento de la vista recibe la configuración del laboratorio con la descripción de las variables compartidas por el laboratorio y construye, para cada una de ellas, un conjunto de campos numéricos que muestran el valor de la variable en el servidor y su homólogo en la aplicación de EJsS.

Este último elemento gráfico está, a su vez, relacionado con la sección *parameters* del fichero de configuración *LabConfig.js* del servidor, ya que contiene la descripción que utilizará el elemento de la vista *LabFunctionParameter* para la generación automática de la interfaz. En concreto, contiene uno o más pares clave-valor, donde la clave es un identificador único y el valor es un objeto con el formato que se describe a continuación:

```
{
    parameter_names: ['Param. 1', 'Param. 2',..., 'Param. N'],
    options: [{
        name: 'Grupo 1',
        parameter_indexes: [],
    },
    ...
    {
        name: 'Grupo M',
        parameter_indexes: [],
    }]
}
```

El campo *parameter_names* contiene una lista de nombres de N parámetros (variables) del servidor, y el campo *options* contiene la definición de M grupos de selectores. La interfaz se genera de manera que se crea un cuadro desplegable con M opciones, tantas como grupos, y en cada una de ellas se mostrará un subconjunto de los parámetros, definido por los respectivos campos *parameter_names*.

A modo de ejemplo, se proporciona una posible configuración:

```
  parameters: {
    reference: {
      parameter_names: ['Amplitud', 'Periodo', 'Offset Y', 'Offset T', 'Tipo'],
      options: [{
        name: 'Senoidal',
        parameter_indexes: [0, 1, 2, 3]
      }, {
```

```
        name: 'Cuadrada',
        parameter_indexes: [0, 1, 2, 3]
      }, {
        name: 'Triangular',
        parameter_indexes: [0, 1, 2, 3]
      }]
    },
    controller: {
      parameter_names: ['Kp / b0', 'Ki / b1', 'Kd / b2', 'N / a1', 'NS / a2'],
      options: [{
        name: 'Lazo Abierto',
        parameter_indexes: [],
      }, {
        name: 'PID',
        parameter_indexes: [0, 1, 2, 3, 4],
      }, {
        name: 'IIR',
        parameter_indexes: [0, 1, 2, 3, 4],
      }]
    }
  },
}
```

En este caso, en el archivo de configuración se definen dos conjuntos de parámetros: *reference* y *controller*. Estos se pueden referenciar directamente desde el elemento *LabFunctionParameter*. Tal y como se muestra en la figura B.5, el campo lab (resaltado en rojo) contiene el nombre de la instancia de laboratorio, mientras que el campo *config* contiene la referencia a la configuración (resaltado en azul), en este caso *reference*. Aunque cada instancia de *LabFunctionParameter* se asocia con un único conjunto de parámetros (un identificador), es posible definir tantas instancias como sea necesario. En esencia, el uso de estos elementos de la vista implica que con poco esfuerzo de desarrollo es posible tener un laboratorio completamente funcional.

Finalmente, una vez que el laboratorio se encuentra listo para su uso, se puede hacer el despliegue directamente desde EJsS, con el botón que el plugin ReNoLabs añade en la botonera principal para tal fin.

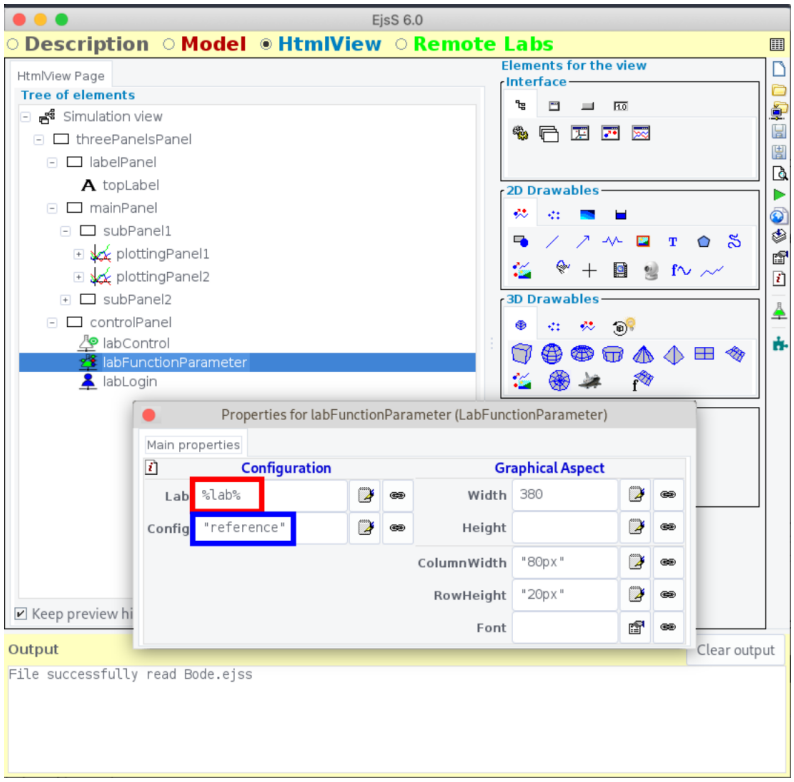


Figura B.5: Configuración del elemento de la vista *LabFunctionParameter*

Apéndice C

Acceso a las prácticas remotas de las asignaturas de Ingeniería de Sistemas y Automática

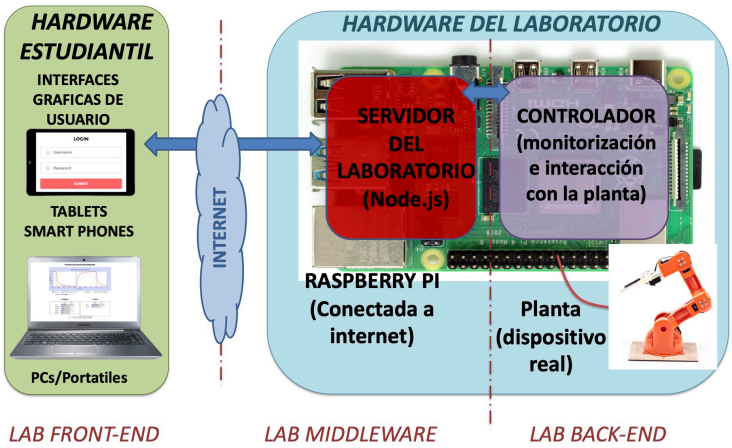
Se han preparado un conjunto de transparencias, con instrucciones genéricas del funcionamiento del laboratorio desde el punto de vista del alumno, para que los profesores puedan adaptarlas a las diferentes experiencias que están realizando.

Estas transparencias están disponibles, en formato powerpoint, en el repositorio en el que se recoge todo el software desarrollado en este PIMCD.

Además, cabe indicar que han sido ya adaptadas, y ampliadas convenientemente, para los trabajos de evaluación continua que se recogen en los apéndices F y H.

Laboratorios Remoto de Ingeniería de Sistemas y Automática

Los laboratorio remotos de Ingeniería de Sistemas y Automática permiten a los alumnos acceder desde su casa a diferentes dispositivos reales e interactuar con ellos (eligiendo que señales se aplican sobre ellos y observando su respuesta por pantalla y a través de diferentes ficheros)
Los elementos que dan soporte al laboratorio se muestran en el siguiente esquema, transparente para los alumnos, que acceden a él a través de una página web tal y como se explicará en las siguientes transparencias



Acceso a los Laboratorios Remotos

Acceder a la página web de las practicas remotas, escribiendo en la URL, la IP indicada en cada asignatura y experimento.

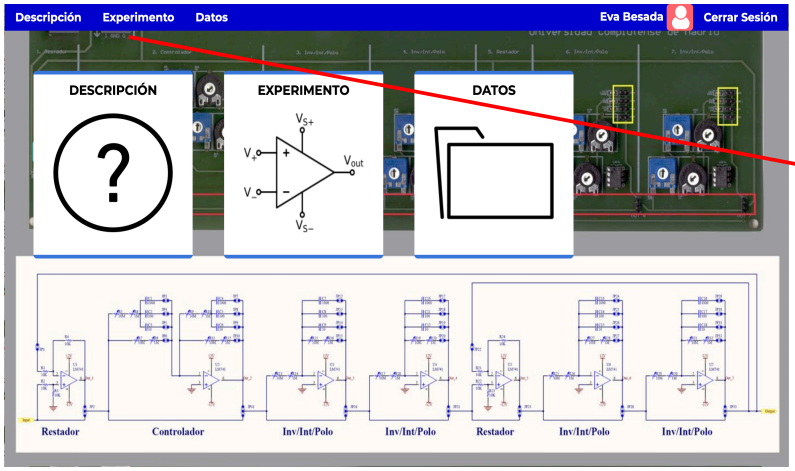
Ejemplo:

147.96.71.236

Acto seguido, hay que introducir el usuario y la password que se le indica a cada alumno

Se puede acceder a las prácticas remotas desde diferentes navegadores de tablets, móviles y PCs. Sin embargo, es preferible hacerlo desde un PC y Firefox.

Acceso a los Laboratorios Remotos



Terminar la sesión

Menú para navegar entre las opciones desde las distintas páginas web del laboratorio

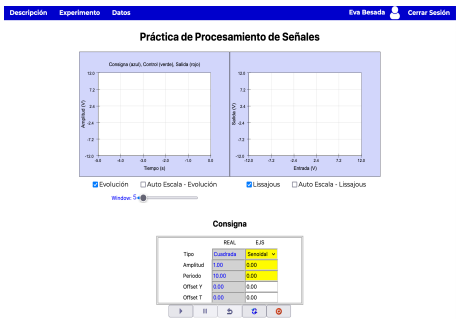
A continuación, se muestra la página principal del laboratorio con iconos que enlazan a diferentes elementos:

- Descripción: información de la práctica
- Experimento: acceso al interfaz de la experiencia
- Datos: acceso a ficheros con resultados

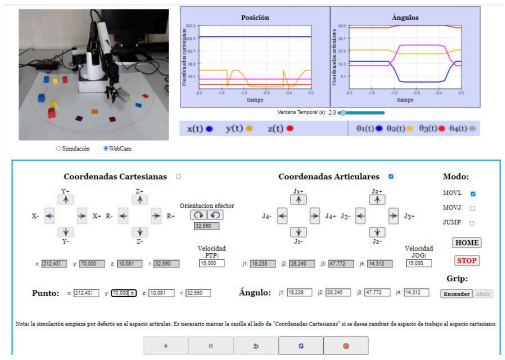
Acceso a los Laboratorios Remotos: Descripción

La descripción recoge información general del laboratorio y particular de la experiencia que se debe realizar.

Acceso a los Laboratorios Remotos: Experimento



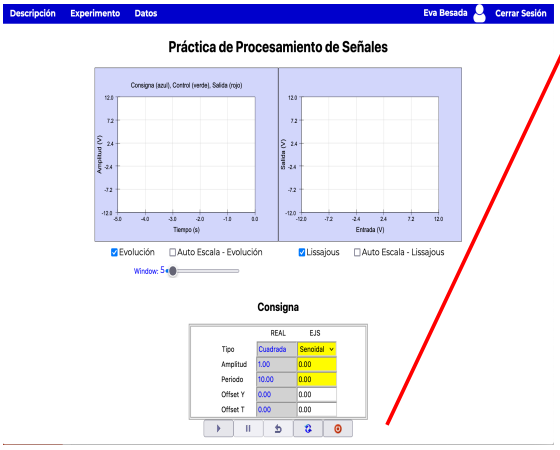
Ejemplo interfaz de Procesamiento de señales



Ejemplo interfaz de Robótica

Tras pinchar el icono EXPERIMENTO se accede a la página que permite interactuar con el dispositivo físico.
Según la experiencia, se mostrarán diferentes tipos de gráficas con la evolución de señales relevantes y una imagen en tiempo real de los dispositivos involucrados.
Además, existirán diferentes tipos de elementos para enviar diferentes señales/órdenes al dispositivo

Acceso a los Laboratorios Remotos: Experimento



En todos los laboratorios, hay un menú de botones inferior, sirve para interactuar con el experimento

Lanzar el experimento


Parar el experimento


Reiniciar el experimento

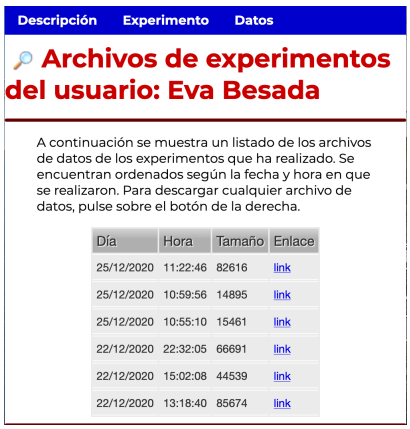

Modificar parámetros de señales/órdenes de entrada


Conectar/desconectarse del experimento

Tras entrar en la página,

1. Pulsar botón de inicio
2. En cualquier momento, modificar valores de los comandos de control y pulsar  para aplicarlos sobre el dispositivo real

Acceso a los Laboratorios Remotos: Datos



Día	Hora	Tamaño	Enlace
25/12/2020	11:22:46	82616	link
25/12/2020	10:59:56	14895	link
25/12/2020	10:55:10	15461	link
22/12/2020	22:32:05	66691	link
22/12/2020	15:02:08	44539	link
22/12/2020	13:18:40	85674	link

Tras pinchar el menu DATOS se observan todos los ficheros que se han almacenado en el servidor para cada alumno, ordenados por fecha y hora.

El alumno puede descargar los ficheros en su PC, tablet o movil pulsando el enlace correspondiente.

El contenido de los ficheros (una matriz con una fila con instante de tiempo y una columna diferente para cada variable registrada) depende de la experiencia. Por lo tanto, el alumno debe revisar el guión de la experiencia o la página de descripción del laboratorio, para poder interpretar la información de cada columna de forma adecuada.

Apéndice D

Información del circuito analógico reconfigurable

El sistema analógico con el que se trabajará en alguna de las prácticas del laboratorio es un circuito reconfigurable, desarrollado en el PIMCD-2006/07-650 [18], formado por un conjunto de amplificadores operacionales, resistencias variables, condensadores y jumpers, en el que se pueden incluir/excluir diferentes elementos y modificar el valor nominal de parte de ellos. La forma en la que ha sido diseñado hace posible que el circuito implemente varios tipos de controladores y sistemas, y por lo tanto, permite realizar diferentes prácticas experimentales, sobre sistemas reales, de las asignaturas de Sistemas Lineales, Procesamiento de Señales, y Control de Sistemas del área de Ingeniería de Sistemas y Automática.

En este apéndice se describe el circuito con el mismo grado de detalle que se les presenta a los alumnos en un manual del laboratorio, ya que resulta de gran utilidad que los alumnos entiendan las posibilidades que éste ofrece para llevar a cabo diferentes prácticas en las asignaturas anteriormente mencionadas.

D.1. Descripción general del circuito analógico

En la parte superior de la figura D.1 se presenta este sistema, llamado Circuito Analógico (Planta + Controlador Continuo). En su parte inferior se muestra el esquemático correspondiente al sistema superior.

El circuito está formado por un conjunto de elementos básicos (4 Inv/Int/Polos, 2 Restadores y 1 Controlador) dispuestos en serie, que pueden ser incluidos o excluidos de los experimentos por medio de la fila de jumpers enmarcados en rojo en la figura. Todos los jumpers de esta fila tienen que estar colocados, bien en su posición inferior (caso en el que que exluirá al elemento colocado antes del jumper del circuito utilizado) o en su posición superior (caso en el que se incluirá el elemento, en serie con el resto de los elementos elegidos).

Cada uno de los elementos básicos está formado a su vez por un conjunto de jumpers de configuración (enmarcados en amarillo y azul en la figura) y, opcionalmente, un conjunto de resistencias variables y condesandores de tres valores diferentes (1000, 100 y 10 nF). El comportamiento de los mismos se detalla en los diferentes apartados de este apéndice.

Para permitir una ajuste fino de los valores de las resistencias de los elementos básicos se han colocado en serie dos potenciómetros variables. Para disminuir el valor total asociado a la pareja de resistencias se girarán los potenciometros hacia la izquierda y para aumentarlos hacia la derecha. Para fijarlo a un valor dado, es necesario eliminar los jumpers del circuito y utilizar un multímetro para medir, en cada par de resistencias los valores entre la patita izquierda de la resistencia de la

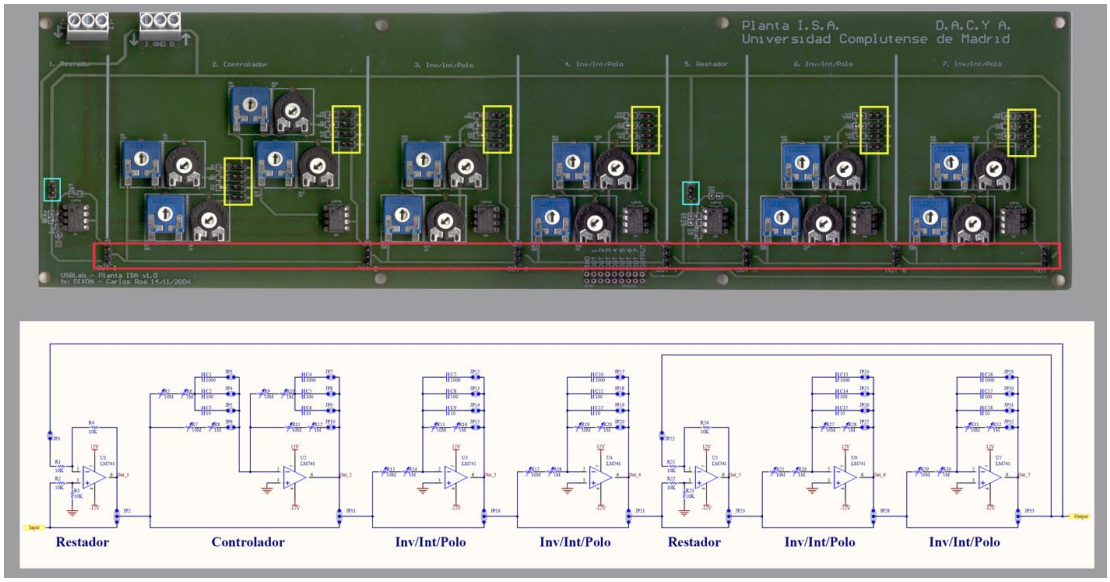


Figura D.1: Circuito analógico reconfigurable

izquierda y la patita central de la resistencia de la derecha. Los valores posibles se encuentran entre 0 ohmnios (los dos potenciometros totalmente girados a la izquierda) y 11 megaohmnios (los dos potenciometros totalmente girados a la derecha).

De los tres condensadores disponibles en cada elemento del circuito sólo se incluirá uno en aquellas configuraciones en las que sea necesario un condensador. Los valores elegidos, de 1000, 100 y 10 nF, permiten modificar rápidamente las características de cada elemento multiplicando o dividiendo por potencias de 10 la disposición de los polos o constantes de integración de los elementos básicos.

Para conectar el circuito a cualquier otro dispositivo, se utilizarán las clemas colocadas en su parte superior izquierda. La de la izquierda sirve para alimentar los amplificadores operacionales del circuito, por lo que será conectada, de izquierda a derecha, a +12V, 0V y -12V. La de la derecha da acceso a las señales de entrada (input) al circuito y a su salida (output) por medio de sus conexiones izquierda y derecha respectivamente.

Además, para que se pueda observar la salida de cualquiera de los elementos básicos incorporados al circuito, se ha dispuesto en su parte inferior central una ristra de pines que dan acceso a las salidas de cada uno de los elementos básicos.

A continuación, en las subsecciones sucesivas, se describe cada uno de los elementos básicos disponibles, indicando como configurarlo correspondiente en sus diferentes variantes y sus funciones de transferencia correspondientes.

D.2. Inv/Int/Polo

Los componentes del circuito que constituyen el elemento básico Inv/Int/Polo permiten que éste sea configurado como una ganancia inversora (Inv), un integrador inversor (Int) o un polo con una ganancia negativa (Polo).

El circuito, representado en la figura D.2, está formado por un amplificador operacional, la resistencia **R1** (compuesta por dos resistencias variables en serie) y, opcionalmente, la resistencia **R2** (compuesta también por dos resistencias variables en serie y que se incluye utilizando su jumper

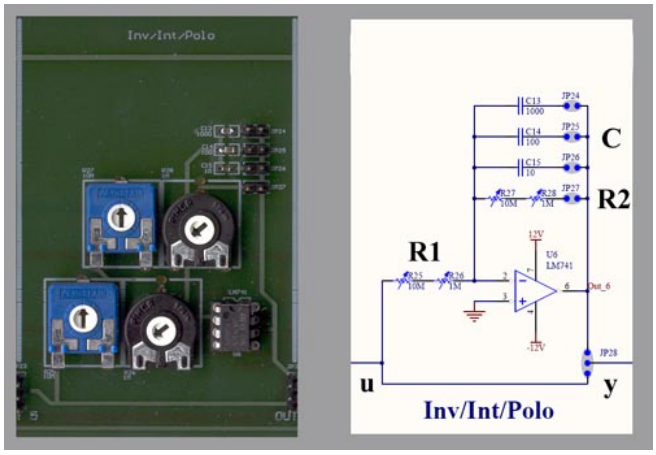


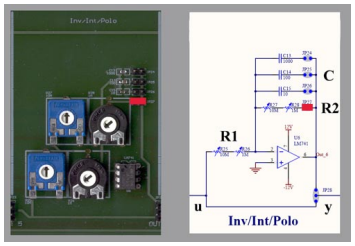
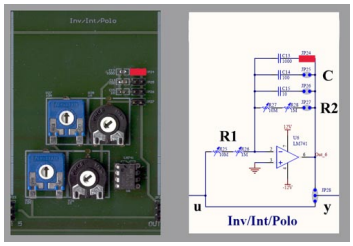
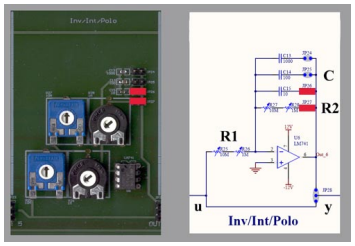
Figura D.2: Inv/Int/Polo

correspondiente) y el condensador **C** (cuyo valor e inclusión en el circuito se fija conectando al circuito por medio de un jumper uno de los tres condensadores posibles).

El comportamiento de la ganancia inversora se consigue incluyendo la resistencia **R2** en el circuito y excluyendo los condensadores. La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R1** y **R2**), y la disposición de los jumpers dentro del elemento, se representan en la columna izquierda de la siguiente tabla. El valor de las resistencias **R1** y **R2** se fijará por medio de sus resistencias variables.

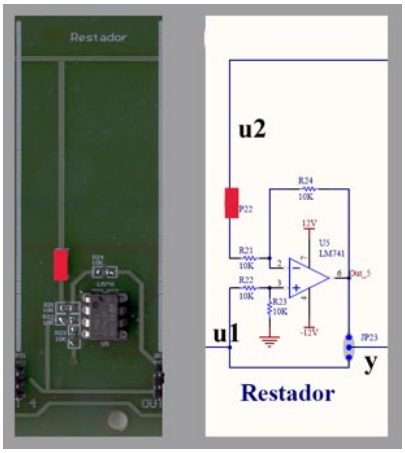
El comportamiento del integrador inversor se consigue excluyendo la resistencia **R2** en el circuito e incluyendo uno de los condensadores **C**. La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R1** y **C**), y la disposición de los jumpers dentro del elemento, se representan en la columna central de la siguiente tabla. En el ejemplo propuesto el valor de **C** será 1000 nF, ya que ese es el valor del condensador incluido por medio del jumper, y el valor de la resistencia **R1** se fijará por medio de sus resistencias variables.

El comportamiento del polo simple con ganancia negativa se consigue incluyendo la resistencia **R2** en el circuito e incluyendo uno de los condensadores **C**. La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R1**, **R2** y **C**), y la disposición de los jumpers dentro del elemento, se representan en la columna derecha de la siguiente tabla. En el ejemplo propuesto el valor de **C** será 10 nF, ya que ese es el valor del condensador incluido por medio del jumper, y el valor de las resistencias **R1** y **R2** se fijará por medio de sus resistencias variables.

Inversor	Integrador	Polo simple
$G(s) = -\frac{R_2}{R_1}$	$G(s) = -\frac{1}{R_1 C s}$	$G(s) = -\frac{R_2}{R_1}\frac{1}{R_2 C s+1}$
		

D.3. Restador

Este elemento, representado en la figura de la tabla siguiente, esta formado por un amplificador, un conjunto de resistencias fijas y un jumper, que tiene que estar colocado siempre y cuando se desee incluir este elemento básico dentro del circuito total con el que se realizarán las prácticas. La relación entre la salida (**y**) y las dos entradas (**u1** y **u2**) es la recogida en la expresión de la tabla siguiente.

$Y(s) = U_1(s) - U_2(s)$


D.4. Controlador

Los componentes del circuito que constituyen el elemento básico Controlador permiten que este sea configurado como un controlador P, I, PI, PD, PID, red adelanto, red atraso, o red de atraso y adelanto. En todos los casos, la función de transferencia del controlador tiene una ganancia negativa,

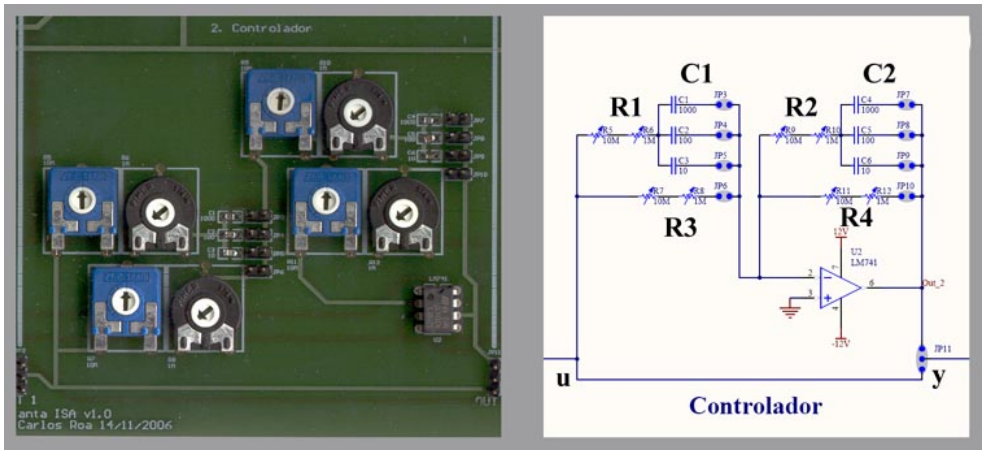


Figura D.3: Controladores reconfigurables

que tendrá que ser compensada, de forma correcta, mediante otros elementos básicos del circuito.

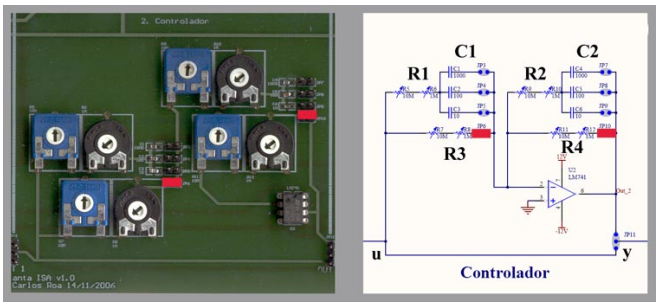
El circuito, representado en la figura D.3, está formado por un amplificador operacional y opcionalmente, las resistencias **R1**, **R2**, **R3** y **R4** (compuesta cada una de ellas por dos resistencias variables en serie) y los condensador **C1** y **C2** (cuyos valores e inclusiones en el circuito se fijan conectando al circuito por medio de un jumper uno de los tres condensadores posibles en cada caso).

D.4.1. Controlador proporcional (P)

El comportamiento de este controlador se consigue:

- incluyendo las resistencias **R3** y **R4** en el circuito
- excluyendo las resistencias **R1** y **R2**, y los condensadores **C1** y **C2**.

La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R3** y **R4**), y la disposición de los jumpers dentro del elemento, se representan en la siguiente tabla. Los valores de las resistencias **R3** y **R4** se fijarán por medio de sus resistencias variables.

$G(s) = -\frac{R_4}{R_3}$


Antes de proseguir, conviene indicar que el comportamiento de este controlador también puede ser alcanzado utilizando cualquiera de los elementos básicos Int/Inv/Polo en su configuración Inversora.

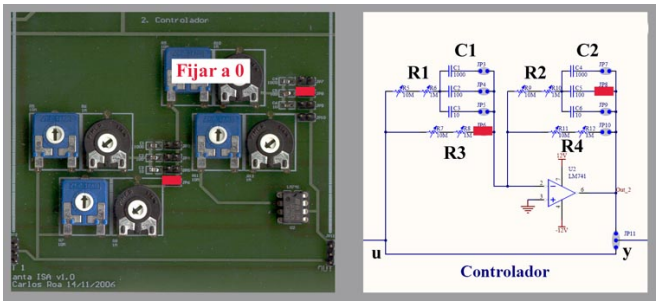
D.4.2. Controlador integral (I)

El comportamiento de este controlador se consigue:

- incluyendo las resistencias **R3** y **R2**, y uno de los condensadores **C2** en el circuito
- excluyendo las resistencias **R1** y **R4**, y los condensadores **C1**.
- fijando a cero (girando los potenciómetros totalmente a la izquierda) el valor de **R2**

La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R3** y **C2**), y la disposición de los jumpers dentro del elemento, se representan en la siguiente tabla. En el ejemplo propuesto el valor de **C2** será 100 nF, ya que ese es el valor del condensador incluido por medio del jumper, y el valor de la resistencia **R3** se fijará por medio de sus resistencias variables.

$$G(s) = -\frac{1}{C_2 R_3 s}$$



Antes de proseguir, conviene indicar también que el comportamiento de este controlador puede ser alcanzado utilizando cualquiera de los elementos básicos Int/Inv/Polo en su configuración Integradora.

D.4.3. Controlador proporcional y derivativo (PD)

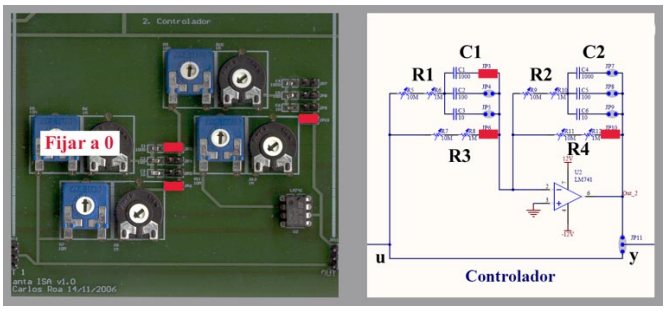
El comportamiento de este controlador se consigue:

- incluyendo las resistencias **R1**, **R3** y **R4**, y uno de los condensadores **C1** en el circuito
- excluyendo la resistencia **R2** y los condersadores **C2**.
- fijando a cero (girando los potenciometros totalmente a la izquierda) el valor de **R1**

La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R3**, **R4** y **C1**), y la disposición de los jumpers dentro del elemento, se representan en la siguiente tabla. En el ejemplo propuesto el valor de **C1** será 1000 nF, ya que ese es el valor del

condensador incluido por medio del jumper, y los valores de las resistencias **R3** y **R4** se fijarán por medio de sus resistencias variables.

$$G(s) = -\frac{R_4}{R_3}(R_3C_1s + 1)$$



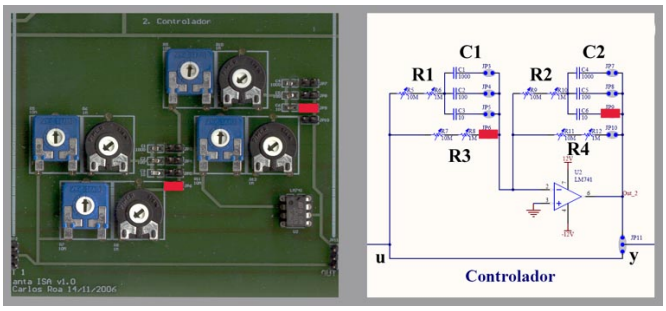
D.4.4. Controlador proporcional e integral (PI)

El comportamiento de este controlador se consigue:

- incluyendo las resistencias **R2** y **R3**, y uno de los condensadores **C2** en el circuito
- excluyendo las resistencias **R1** y **R4** y los condensadores **C1**.

La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R2**, **R3** y **C2**), y la disposición de los jumpers dentro del elemento, se representan en la siguiente tabla. En el ejemplo propuesto el valor de **C2** será 10 nF, ya que ese es el valor del condensador incluido por medio del jumper, y los valores de las resistencias **R2** y **R3** se fijarán por medio de sus resistencias variables.

$$G(s) = -\frac{R_2C_2s+1}{R_3C_2s}$$



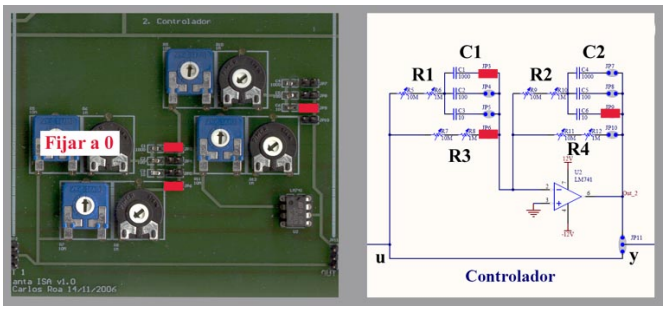
D.4.5. Controlador proporcional, integral y derivativo (PID)

El comportamiento de este controlador se consigue:

- incluyendo las resistencias **R1**, **R2** y **R3**, y uno de los condensadores de **C1** y **C2** en el circuito
- excluyendo la resistencia **R4**.
- fijando a cero (girando los potenciómetros totalmente a la izquierda) el valor de **R1**

La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R2**, **R3**, **C1** y **C2**), y la disposición de los jumpers dentro del elemento, se representan en la siguiente tabla. En el ejemplo propuesto el valor de **C1** será 1000 nF y el valor de **C2** 10 nF, ya que ese es el valor de los condensadores incluidos por medio del jumper, y los valores de las resistencias **R2** y **R3** se fijarán por medio de sus resistencias variables.

$$G(s) = -\frac{(R_3C_1s+1)(R_2C_2s+1)}{R_3C_2s}$$



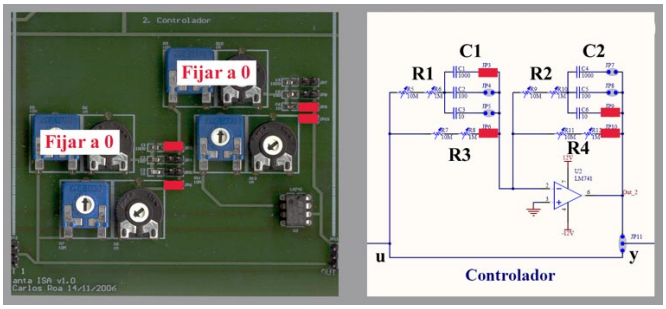
D.4.6. Red de adelanto o retraso

El comportamiento de este controlador se consigue:

- incluyendo las resistencias **R1**, **R2**, **R3** y **R4**, y uno de los condensadores de **C1** y **C2** en el circuito
- fijando a cero (girando los potenciometros totalmente a la izquierda) el valor de **R1** y **R2**

La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R3**, **R4**, **C1** y **C2**), y la disposición de los jumpers dentro del elemento, se representan en la siguiente tabla. En el ejemplo propuesto el valor de **C1** será 1000 n y el valor de **C2** 10 nF, ya que ese es el valor de los condensadores incluidos por medio del jumper, y los valores de las resistencias R3 y R4 se fijarán por medio de sus resistencias variables.

$$G(s) = -\frac{R_4}{R_3}\frac{R_3C_1s+1}{R_4C_2s+1}$$



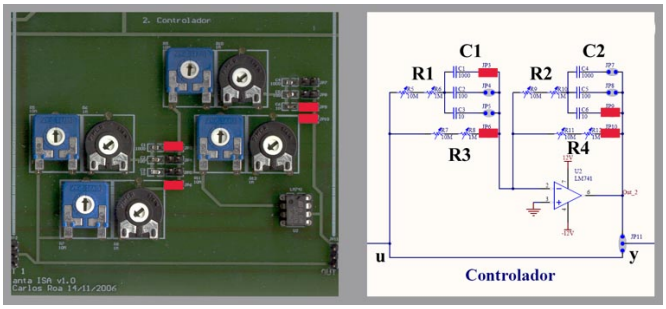
D.4.7. Red de adelanto y retraso

El comportamiento de este controlador se consigue:

- incluyendo las resistencias **R1**, **R2**, **R3** y **R4**, y uno de los condensadores de **C1** y **C2** en el circuito

La expresión que relaciona la entrada (**u**) y la salida (**y**) con los valores de los componentes utilizados (**R1**, **R2**, **R3**, **R4**, **C1** y **C2**), y la disposición de los jumpers dentro del elemento, se representan en la siguiente tabla. En el ejemplo propuesto el valor de **C1** será 1000 nF y el valor de **C2** 10 nF, ya que ese es el valor de los condensadores incluidos por medio del jumper, y los valores de las resistencias **R1**, **R2**, **R3** y **R4** se fijarán por medio de sus resistencias variables.

$$G(s) = -\frac{R_4}{R_3}\frac{([R_1+R_3]C_1s+1)(R_2C_2s+1)}{(R_1C_1s+1)([R_2+R_4]C_2s+1)}$$



D.5. Observaciones generales

- La alimentación elegida para los operacionales del circuito planta recortará las señales de entrada a los mismos superiores a +12V e inferiores a -12V. Por lo tanto, para conseguir un

comportamiento lineal del sistema formado por los diferentes elementos básicos del circuito hay que asegurar que ninguna de las etapas alcanza los limites de saturación.

- Ante cualquier duda sobre las conexiones finales de los circuitos, el alumno debe siempre consultar al profesor o al técnico del laboratorio.
- Cuando el sistema no funciona o la respuesta del mismo no es la esperada, se debe:
  1. Comprobar que todos los elementos hardware están conectados correctamente y que la alimentación está encendida.
  2. Comprobar uno a uno el correcto funcionamiento de los elementos básicos hasta detectar el elemento erróneo, e indicárselo al profesor para que el defecto sea corregido por los técnicos del laboratorio.
  3. Disminuir el valor de la señal de entrada, ya que puede que se esté saturando alguna de las etapas del circuito o que las señales de salida se encuentre fuera del rango de funcionamiento de la tarjeta de adquisición de datos.
  4. Llamar al profesor, si las comprobaciones anteriores no resuelven el problema.

Apéndice E

El circuito de adecuación de señales entre la Raspberry PI y el circuito analógico reconfigurable

Para conectar los pines de entrada y salida de la Raspberry PI a las señales de entrada y salida del circuito analógico reconfigurable es necesario que las señales de la Raspberry PI se adecuen a los rangos adecuados del circuito. Para ello, se ha diseñado he introducido, entre ambas, tal y como se muestra en la Figura E.1, el circuito de adecuación de señales que se muestra en las sucesivas figuras de este apéndice.

La figura E.2 muestra en detalle la placa de inserción empleada para implementar el circuito de adecuación que sirve de enlace entre la planta (circuito analógico) y la Raspberry PI. Sus elementos fundamentales son el conversor de digital a analógico (DAC) MCP4812 (fabricado por Microchip) y el circuito integrado TL84C (fabricado por Motorola) que contiene cuatro amplificadores operacionales.

A continuación se describen con más detalle cada componente y su utilidad en la placa de adecuación de señales.

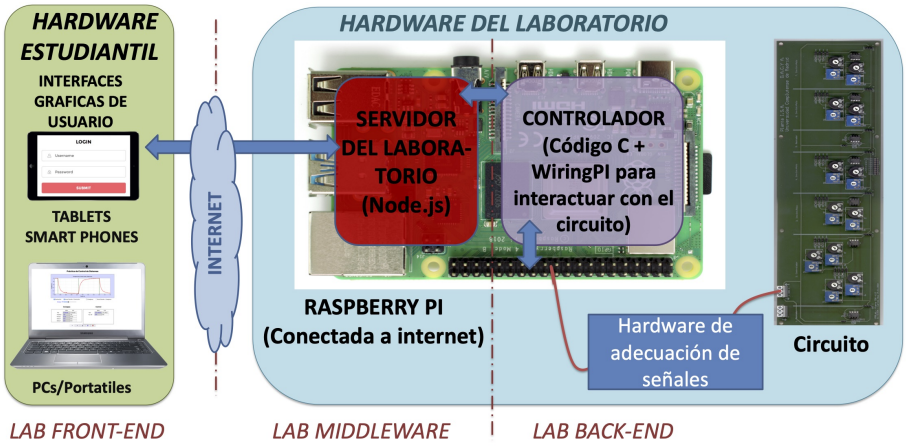


Figura E.1: Vista general de los elementos de los laboratorios remotos sobre circuitos

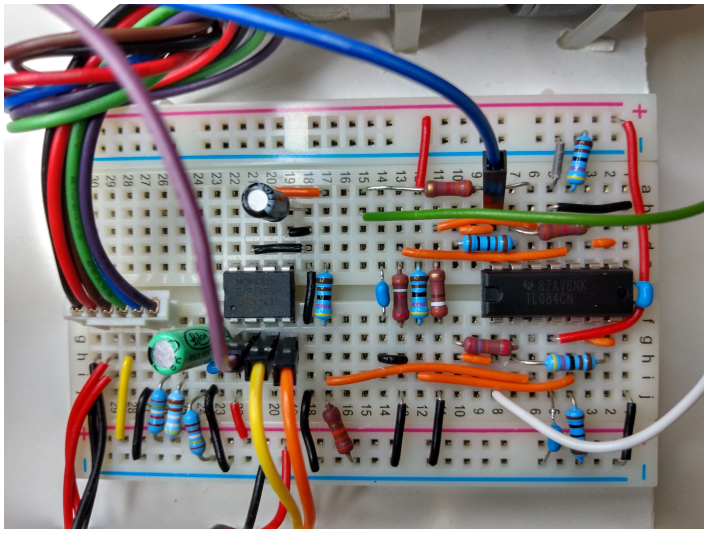
t! 

Figura E.2: Vista de la placa de entrada/salida

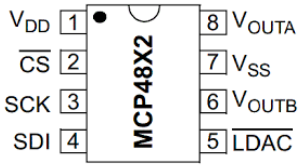


Figura E.3: *Pin-out* del DAC MCP4812

E.1. Los conversores digital-analógico y analógico-digital

En esta sección se indican los conversores utilizados para convertir las señales digitales de la Raspberry PI en las señales analógicas necesarias para el circuitos

E.1.1. El conversor digital-analógico

Como la Raspberry PI no dispone de un conversor digital-analógico propio para generar el voltaje de entrada al circuito analógico configurable, es necesario incluir un DAC en la placa de adecuación de señales. Con este objetivo se ha elegido el **MCP4812**, un conversor DAC de 10 bits. Este chip, cuyo esquema de conexiones se muestra en la figura E.2, se conecta como esclavo mediante un protocolo de comunicación SPI a un maestro (en nuestro caso la Raspeberry PI) que le debe suministrar una secuencia de 16 bits.

Además, es interesante hacer notar que el dispositivo no emplea toda la potencialidad del interfaz SPI. De hecho, el dispositivo **MCP4812** recibe mensajes a través de la conexión MOSI, pero no envía mensajes a través la conexión MISO.

A continuación analicemos el *pin-out* del chip y sus funciones (los números de la descripción se corresponden al número del pin correspondiente, apreciable en la figura E.3).

1. V_{DD}: voltaje de alimentación del DAC. Como admite un rango de 2,7 a 5,5 V, será conectado al voltaje de alimentación proporcionado por la Raspberry PI.

2. $\overline{\text{CS}}$: pin de entrada utilizado para selección el dispositivo mediante el interfaz SPI. En el caso de nuestro dispositivo, para que este dispositivo se encuentre seleccionado debe recibir un 0 lógico. Con este fin, será conectado a una señal digital de salida de la Raspberry PI.

3. SCK: pin de entrada de la señal de reloj que sincroniza la comunicación entre este dispositivo y el máster. Por lo tanto, será conectado a la salida del SCK del SPI de la Raspberry PI.

4. SDI: entrada para recibir los datos enviados por el máster. Corresponde a la conexión MISO de la interfaz SPI de la Raspberry PI.

5. $\overline{\text{LDAC}}$: permite controlar el momento en que se realiza la conversión, es decir, en el que la salida del DAC toma el valor del voltaje correspondiente a los valores de la entrada digital que el DAC ha recibido por el SDI. En nuestro caso lo mantendremos siempre conectado a tierra, de forma que la conversión se realice de forma inmediata, tan pronto como la señal digital haya sido recibida.

6. V_{OUTB}: salida analógica B (de las dos salidas analógicas disponibles en el chip). En nuestro caso, será la salida utilizada para generar el voltaje que, una vez conformado por la etapa de adecuación de la señal de salida, será utilizado como entrada de la planta (el circuito analógico reconfigurable). Por lo tanto, conectaremos esta salida a una de las entradas de los amplificadores de adecuación de señales.

7. V_{ss}: conexión a tierra del chip (que será unida a la tierra de la Raspberry PI y del circuito analógico de la planta).

8. V_{OUTA}: salida analógica A, que no será empleada en esta práctica.

Tal y como se ha mencionado, el DAC recibe a través de la entrada SDI secuencias de 16 bits, que son almacenadas en un registro de entrada. Los cuatro bits más significativos del mensaje (o registro) se emplean para configurar el dispositivo, los 10 siguientes contienen el valor del voltaje de salida que se ha de transferir a través de la salida analógica del DAC, y los dos últimos bits, en el caso de nuestro conversor, se desprecian. Mas en concreto, la estructura del mensaje (y del registro correspondiente) es la siguiente:

bit 15	Permite seleccionar la salida del conversor que se va a emplear 1 = Salida V_{OUTB} 0 = Salida V_{OUTA}
bit 14	No se emplea
bit 13	Bit para seleccionar la ganancia de la salida 1 = 1X, $\text{V}_{\text{OUT}} = \text{V}_{\text{REF}} \cdot \text{D}/4096$ 0 = 2X, $\text{V}_{\text{OUT}} = 2 \cdot \text{V}_{\text{REF}} \cdot \text{D}/4096$ donde D son los 10 bits con los datos de entrada $V_{REF} = 2{,}048V$ el voltaje de referencia del DAC.
bit 12	Bit de activación de la salida 1 = La salida está activada 0 = La salida está desactivada
bits 11-0	Bits de datos de entrada (D) al DAC bits 11-2 asociados a los 10 bits del DAC bits 1-0 valores ignorados (se prefijan 0)

Es importante aclarar que aunque es un conversor DAC de 10 bits, el valor (D) de los datos de entrada, tal y como se muestra en la tabla anterior, se representa con 12 bits (estando sus

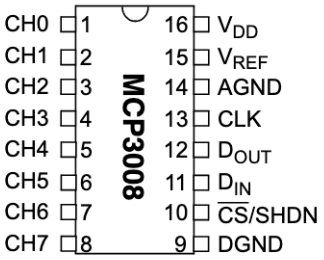


Figura E.4: *Pin-out* del ADC MCP3008

dos últimos fijos a 0). Esto hace que el valor real (D) representado está siempre desplazado dos posiciones a la izquierda: D = Valor 10 bits $\cdot\, 2^2$, y que su valor mínimo sea igual a 0 y su valor máximo igual a 4096.

Además, como trabajaremos con ganancia 2X (poniendo el bit 13 = 0) y este DAC tiene definido un voltaje de referencia $\mathrm{V_{ref}} = 2{,}048V$, el valor mínimo del voltaje de salida será igual a cero y el máximo, cuando los 10 primeros bits de D toman valor 1 (lo que equivale a 1023 en decimal) y los dos restantes 0, será igual a:

$$\mathrm{V_{OUT}^{MAX}} = 2 \cdot 2{,}048 \cdot \frac{1023 \cdot 2^2}{4096} = 4{,}092V$$

Podemos así mismo obtener la cuantización del voltaje de salida, es decir, calcular la diferencia entre dos valores de voltaje consecutivos generables por el DAC. Dicha distancia, que recibe el nombre de LSb (Least Significant bit), está relacionada con el tamaño del conversor (en nuestro caso 10 bits). El cálculo de su valor, para nuestro caso, se realiza en la siguiente expresión

$$\mathrm{LSb} = \frac{2 \cdot 2{,}048}{1024} = \frac{4{,}096}{1024} = 4mV$$

Los valores calculados para $\mathrm{V_{OUT}^{MAX}}$ y LSb son, lógicamente, teóricos. En los experimentos reales se ven afectados por el ruido y la precisión del DAC. Para más información, es conveniente consultar el *datasheet* del dispositivo [19].

E.1.2. El conversor analógico-digital

El ADC MCP3008 de Microchip es el dispositivo elegido para convertir la señal proveniente del circuito analógico y adecuada a los niveles correspondientes en una señal digital de 10 bits que es comunicada a la Raspberry PI a través del protocolo de comunicación SPI de ambos dispositivos.

En este caso, al igual que en el del DAC, la Raspberry PI actúa como el master y el ADC MCP3008 como esclavo. Por lo tanto la conexión del ADC, de acuerdo con el esquemático de la Figura E.4 y el datasheet del fabricante [20], es la que se muestran en la Tabla **??**.

E.2. Adecuación de las señales de entrada y salida mediante el dispositivo TL84C

Como se ha descrito en el apéndice D, la placa del circuito analógico reconfigurable (planta + controles analógicos) esta construida empleando amplificadores operacionales y trabaja en un rango de voltajes de entrada y salida de $\pm 12V$. La salida del DAC de la placa de adecuación de

Pin-out del ADC	Conexión/Valor
1	Señal analógica procedente del circuito de adecuación
2	–
3	–
4	–
5	–
6	–
7	–
8	–
9	0V
10	Salida digital de la Raspbery PI, para inicializar cada conversión
11	Entrada para recibir los datos enviados por el máster. Corresponde a la conexión MISO de la interfaz SPI de la Raspberry PI.
12	Salida para enviar los datos de la conversión. Corresponde a la conexión MOSI de la interfaz SPI de la Raspberry PI.
13	Salida del SCK del SPI de la Raspberry PI
14	0V
15	3.3V
16	voltaje de alimentación proporcionado por la Raspberry PI

Cuadro E.1: Tabla con las conexiones del ADC

señales es capaz de suministrar valores en el rango 0 a 4,092V, y la entrada del conversor analógico digital (ADC) es capaz de leer señales en el rango 0 a 3,3V. Por lo tanto, es necesario adecuar los valores de los conversores al rango de funcionamiento del circuito reconfigurable, utilizando para ese fin los circuitos descritos a continuación y el dispositivo TL084C de Motorola.

E.2.1. Adecuación de la salidas del DAC a las entradas del circuito reconfigurable

Para adecuar (reescalar y desplazar) los valores de salida del DAC (0-4.092V) al rango de voltaje de entrada de la placa del circuito reconfigurable ($\pm 12V$) se emplea el circuito de la figura E.5, que está formado por dos etapas diferenciadas:

- La primera etapa del circuito está formada por un divisor de tensión seguido de un amplificador operacional en modo de seguidor de tensión. La salida de esta etapa se puede calcular directamente a partir del divisor de tensión:

$$V_1 = +12 \cdot \frac{R_{b1}}{R_{b1} + R_{b2}}$$

- La segunda etapa es un restador-inversor. La entrada negativa se alimenta mediante la salida de la etapa anterior (V_1). La entrada positiva es directamente la salida elegida del DAC (V_{OUTB}). Los valores de la tensión a la salida de la segunda etapa (V_2), que será conectada a la entrada del circuito analógico reconfigurable, se puede calcular con la siguiente expresión:

$$V_2 = V_{OUTB} \left(\frac{(R_1 + R_3) \cdot R_8}{(R_8 + R_6) \cdot R_3} \right) - V_1 \left(\frac{R_1}{R_3} \right)$$

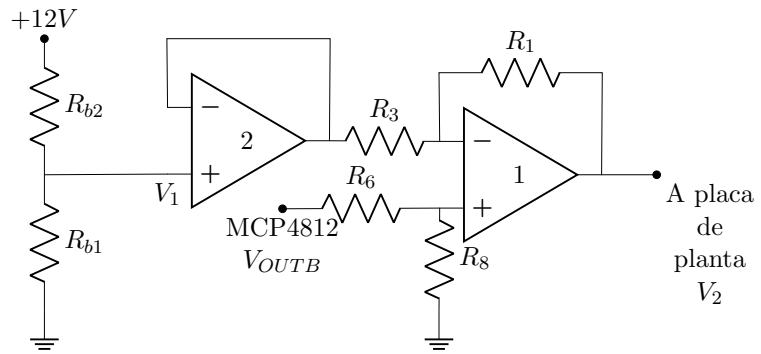


Figura E.5: Adaptador de la salida del DAC (0-4.092V) a los niveles $\pm 12V$ del circuito

La elección de valores de las resistencias del circuito, con el objeto de que la V_2 tome respectivamente los valores de $-12V$ y $12V$ cuando la V_{OUTB} toma los valores de $0V$ y $4{,}096V$, no es única. Para la placa de adecuación de señales, los valores de las resistencias involucradas se han ajustado aproximadamente empleando valores comerciales adecuados. En la siguiente tabla muestra los valores de las resistencias empleadas y la relación entre V_{OUTB} y V_2.

Resistencias circuito de la figura E.5							
R_{b1}	R_{b2}	R_3	R_6	R_1	R_8	V_{OUTB}	V_2
1K	4.7K	4.7K	4.7	27K	33K	$0/+4{,}096$	$-12{,}09/+12{,}06$

E.2.2. Adecuación de la salidas del circuito reconfigurable a la entrada del ADC

Tanto para poder monitorizar la salida del circuito configurable (planta + control continuo), como para poder cerrar un lazo de control con el programa Controlador, necesitamos convertir la salida analógica que nos da el circuito reconfigurable en un una señal digital. Para hacerlo empleamos directamente el conversor ADC MCP3008 de Microchip. Teniendo en cuenta el rango de funcionamiento del circuito reconfigurable y del ADC, debemos convertir los $\pm 12V$ de la salida del primero a los $(0V - 3{,}3V)$ de la entrada del segundo.

Para ello, emplearemos un circuito similar al de caso anterior, con valores adecuados de las resistencias y de la tensión a la que se conecta el divisor de tensiones. No volveremos a repetir el análisis. La figura E.6 muestra el circuito empleado. Las relaciones de voltajes pueden expresarse en este caso como:

$$V_3 = -12 \cdot \frac{R_{a1}}{R_{a1} + R_{a2}}$$

$$V_5 = V_4 \left(\frac{(R_2 + R_4) \cdot R_9}{(R_9 + R_7) \cdot R_4} \right) - V_3 \left(\frac{R_2}{R_4} \right)$$

Para lograr que V_5 tome respectivamente los valores de $0V$ y $3{,}3V$ cuando V_4 toma los valores de $-12V$ y $12V$, se han elegido, siguiendo una metodología similar a la del caso anterior, los valores de las resistencias que se presentan en la siguiente tabla.

Resistencias circuito de la figura E.6							
R_{a1}	R_{a2}	R_4	R_7	R_2	R_9	V_4	V_5
3.9K	4.7K	3.3K	3.3	1K	390Ω	$-12{,}09/+12{,}06$	$-0{,}01/+3{,}31$

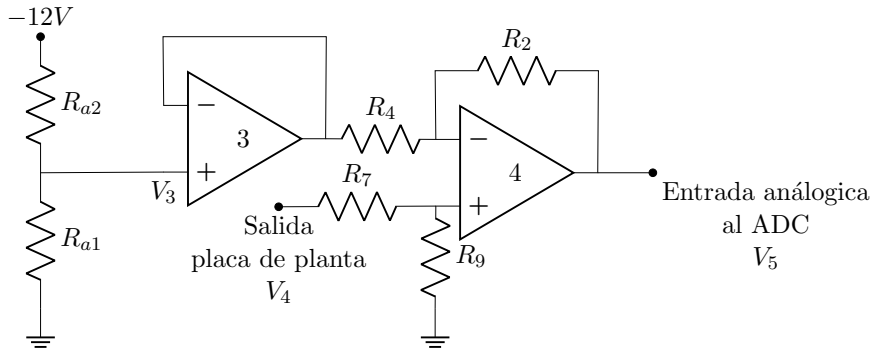


Figura E.6: Adaptador de la salida del circuito analógico ($\pm 12V$) a los niveles del ADC ($0-3{,}3V$)

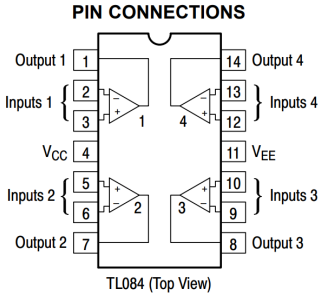


Figura E.7: Circuito integrado TL084C

E.2.3. Conexión del TL84C

Los dos circuitos que acabamos de describir se han implementado empleando un solo chip que dispone de cuatro amplificadores operacionales: el TL084C de la compañia Motorola.

La figura E.7 muestra el esquemático de sus conexiones. En la placa de adecuación de las señales los circuitos se han montado de modo que la numeración de los operacionales coincide con las de las figuras E.5 y E.6. Por lo tanto, las conexiones de sus pines son las que se describen en la segunda columna de la tabla E.2. Para más información, consultar el *datasheet* del dispositivo [21]

E.3. Depuración de la placa de adecuación de señales

Para depurar las posibles causas del mal funcionamiento de la placa de adecuación de señales, se puede utilizar cualquiera de las experiencias de las asignaturas de Sistemas Lineales. Para ello, conectaremos, a través del interfaz gráfico de la experiencia, el valor de la señal de salida V_2 a la entrada V_4 del módulo de entrada y salida, y ejecutamos el modelo durante el tiempo que se desee depurar las señales del circuito.

Para depurar las posibles causas del mal funcionamiento de la placa de adecuación de señales, se puede utilizar cualquiera de las experiencias de las asignaturas de Sistemas Lineales, y aplicarle al circuito, a través del interfaz gráfico de la experiencia, una señal escalón u onda cuadrada de amplitud nula. Además, colocará todos los jumpers del circuito analógico de forma que no se incluya ninguno de sus elementos en el sistema[1].

[1]Es decir, se elimina la influencia del circuito reconfigurable de las conexiones.

Pin	Descripción	Valores de depuración	
		Cte	Variable
1	$V_2 \rightarrow$ Entrada al circuito reconfigurable		0V
2	Entre R_1 y R_3		
3	Entre R_8 y R_6		
4	12 V	12V	
5	V_1, entre R_{b2} y R_{b1}		
6	Al pin 7 de TL084		
7	Izquierda de R_3	2V	
8	Izquierda de R_4	-5.5V	
9	Al pin 8 de TL084		
10	V_3, entre R_{a2} y R_{a1}		
11	-12 V	-12V	
12	Entre R_7 y R_9		
13	Entre R_4 y R_2		
14	V_5, ADC de Microchip		1.6V

Cuadro E.2: Conexiones y valores de depuración del circuito TL084C

Con esa entrada constante:

- La salida del conversor DAC (pin 6 MCP48X2, conectada a la izquierda de la resistencia R6) debe tomar el valor de 2V y la salida V_2 del circuito de la figura E.5 deber si igual a 0V[2].
- La entrada al conversor ADC (pin 14 del TL084C) debe tomar el valor de 1.6 V[3]. La respuesta que se observe en la gráfica de evolución del interfaz de la experiencia debe ser 0[4].

Para facilitar el proceso de depuración, los valores anteriormente mencionados se han representado en la tercera columna de la tabla E.2. Además, también se indican algunos valores constantes, independientes del valor de la señal aplicada por el interfaz gráfico de la experiencia, del circuito de adecuación, que también deben ser revisados (ya que están asociados a la alimentación de los amplificadores operacionales y a las tensiones constantes utilizadas para adecuar las señales a los rangos adecuados).

[2]El programa Controlador convierte el valor de 0V que debe aplicar a la planta al valor medio del rango permitido en el DAC y el circuito de adecuación de la placa de adecuación de señales hace que el valor vuelva a tomar el valor de 0V.

[3]El valor de 0V que tenemos en la entrada de la segunda etapa de adecuación se debe convertir a 1.6V, que es el valor medio del máximo de 3.3V del ADC elegido.

[4]Es decir, igual a la entrada aplicada, ya que se ha conectado directamente la entrada aplicada con la salida del circuito.

Apéndice F

Trabajo de Evaluación Continua de Sistemas Lineales: Identificación de un sistema mediante el análisis de su respuesta temporal

Este trabajo de evaluación continua se lleva realizando, de forma remota, en la asignatura de Sistemas Lineales de la titulación de Ingeniería Electrónica en Comunicaciones desde el curso 2008-2009. Por lo tanto, lo que se ha hecho en este caso es adaptar los diferentes módulos software existentes a la nueva metodología de creación de laboratorios remotos que se ha desarrollado en este PIMCD.

A continuación, se muestra el guión del Trabajo de Evaluación Continua que se les proporciona a los alumnos, actualizado con las modificaciones necesarias para reflejar la forma en la se ha realizado este trabajo desde este curso académico.

F.1. Guión del Trabajo de Evaluación Continua

A continuación se presentan los objetivos y las tareas de identificación que se deben realizar en esta trabajo de evaluación continua (2 sobre los datos de sendos sistemas simulados y otras 2 sobre sendos circuitos analógicos configurados de forma diferente).

F.1.1. Objetivo

El objetivo de este trabajo de evaluación continua es obtener la función de transferencia asociada a los modelos de diferentes sistemas a partir del estudio de su respuesta temporal frente a la entrada escalón. Para alcanzarlo, el alumno: 1) observará la forma de la respuesta, 2) supondrá si se trata de un sistema de primer o segundo orden, 3) obtendrá la función de transferencia a partir de las características observadas y 4) compará de forma gráfica si la respuesta del sistema y la de la función de transferencia propuesta coinciden.

F.1.2. Sistemas simulados

- La respuesta de un sistema continuo a la entrada escalón unitario se encuentra recogida en la variable matricial data almacenada en el fichero Respuesta1.mat. Para analizarla, el alumno ejecutará la orden load Respuesta1.mat y representará los valores de la segunda columna

(salida) frente a los valores de la primera (tiempo). A continuación, tras el analisis de la respuesta temporal, propondrá una función de transferencia de primer o segundo orden para el sistema, y comprobará si la respuesta de dicha función de transferencia coincide con la almacenada en el fichero. De no ser así, buscará otra función de transferencia continua que modele mejor el comportamiento del sistema.

- Repetir la tarea anterior con el sistema discreto cuya respuesta se encuentra almacenada en el fichero Respuesta2.mat

F.1.3. Circuitos reales

- La respuesta de un circuito real formado por amplificadores operacionales, resistencias y condensadores, se puede obtener conectándose a internet con un navegador a la dirección http://XXX.XX.XX.XX siguiendo el procedimiento detallado en la sección F.2. El alumno: 1) observará la respuesta a la entrada escalón seleccionando una entrada tipo Step (o Square con un periodo lo suficientemente largo para que la respuesta pueda alcanzar el estacionario), 2) descargará el fichero generado durante el experimento y lo cargará en Matlab mediante la orden data=load(*NombreFichero*.txt), 3) seleccionará en la variable data el trozo correspondiente al escalón elegido (representando el valor de la segunda columna (entrada) frente a la primera (tiempo)), 4) analizará las características de la respuesta (representando el valor de la cuarta columna (salida) frente a la primera (tiempo)), y 5) propondrá una función de transferencia continua cuya respuesta a la entrada escalón se ajuste al circuito accesible desde internet en la dirección propuesta.

  Ayuda: Un ejemplo de código de Matlab asociado a los pasos pasos 2 y 3 se muestra al final del documento de la sección F.2. Además, es posible que la salida de alguno de los circuitos presente un offset. Para comprobarlo, lo más conveniente es comprobar si la respuesta a la entrada nula es o no nula. De no serlo, corregir el valor de la respuesta con la desviación que hay respecto al valor nulo.

- Repetir la tarea anterior con el circuito accesible a través de la dirección http://XXX.XX.XX.XXX.

Para realizar estas dos últimas tareas el usuario y password de cada alumno es el siguiente:

- Usuario: El mismo que tenéis en vuestra dirección de correo de la UCM. Es decir, si vuestro correo es abcd@ucm.es, el usuario en la página web del laboratorio es abcd.

- El password está formado por tres partes, separadas por un -:

  - Las tres primeras letras del nombre del alumno (en mayúsculas).
  - Los tres primeros número del teléfono consignado en la UCM.
  - Las tres últimas letras del NIF.

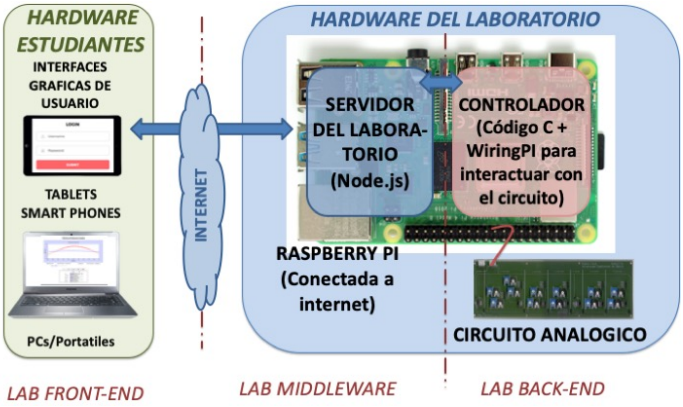
  A modo de ejemplo, si el alumno se llama RAUL, su télefono es 656231427 y su DNI es 03421531X, entonces la password es: RAU-656-31X

F.2. Información adicional del Trabajo de Evaluación Continua

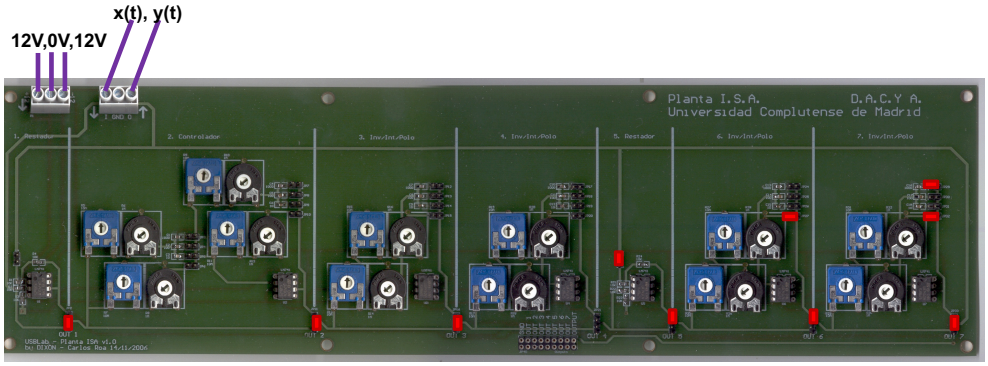
A los alumnos también se les proporciona, de forma adicional, unas transparencias (que se presentan a continuación) para que aprendan a utilizar la página web del laboratorio remoto.

Laboratorio Remoto de Sistemas con Circuitos

El laboratorio remoto de circuitos permite a los alumnos acceder desde su casa a los circuitos reconfigurables del laboratorio e interactuar con ellos (eligiendo que señales se aplican sobre el y observando su respuesta por pantalla y a través de ficheros)
Los elementos que dan soporte al laboratorio se muestran en el siguiente esquema, transparente para los alumnos, que acceden a el a través de una página web tal y como se explicará en las siguientes transparencias



Laboratorio Remoto de Sistemas con Circuitos



Circuito reconfigurable analógico

Es un circuito formado por amplificadores operacionales (741), potenciómetros, condensadores y jumpers (rojo), que según como se configue puede comportarse como un polo simple, doble, complejo conjugado, triple, etc.
Información adicional del circuito, puede verse en la página de ayuda (descripción) del laboratorio remoto (accesible tal y como se muestra a continuación).
En la asignatura de Sistemas Lineales, el alumno utilizará el circuito con la configuración que seleccionen los profesores.

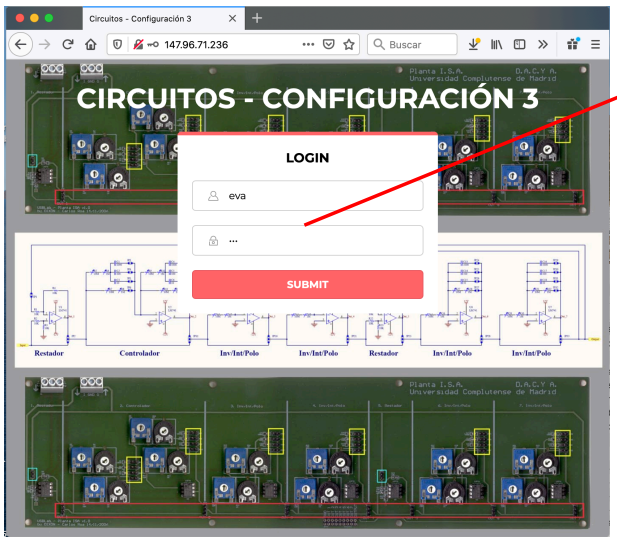
Acceso Laboratorio Remoto: Sistemas con Circuitos

Acceder a la página web de las practicas remotas, escribiendo en la URL, la IP indicada en cada experimento.

Ejemplo:
147.96.71.236

Se puede acceder a las prácticas remotas desde diferentes navegadores de tablets, móviles y PCs. Sin embargo, es preferible hacerlo desde un PC y Firefox.

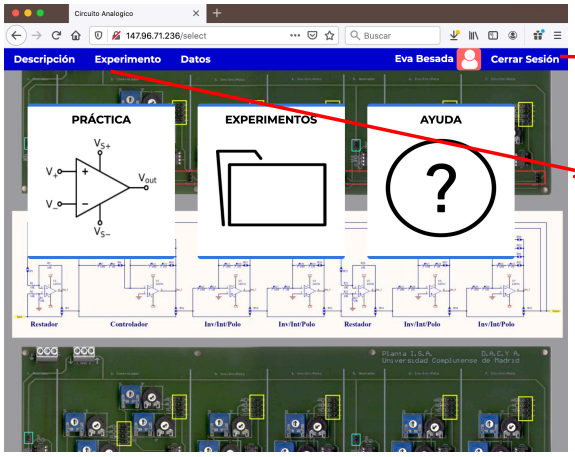
Acceso Laboratorio Remoto: Sistemas con Circuitos



Acto seguido, hay que introducir el usuario y la password del alumno

El alumno utilizará el usuario y password que se le indica en el guión de la práctica. Debe ser utilizado para todas las prácticas y TECs remotas que se realicen durante el curso 2020/21

Acceso Laboratorio Remoto: Sistemas con Circuitos



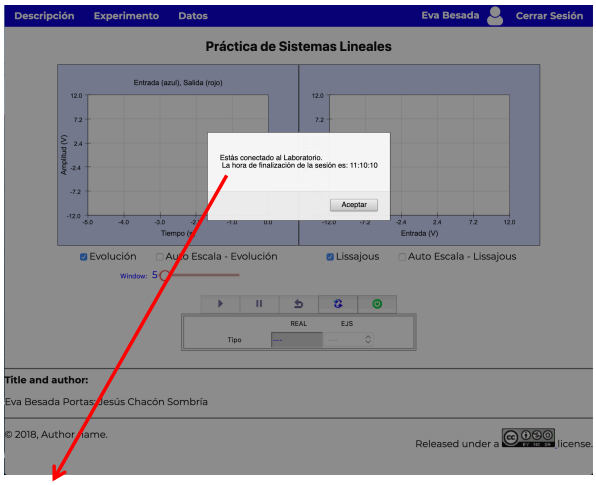
Terminar la sesión

Menú para navegar entre las opciones
Descripción = Ayuda
Experimento = Practica
Datos= Experimentos
(ver descripciones a continuación)

A continuación, se muestra la página principal del laboratorio con iconos que enlazan a diferentes elementos:

- Práctica: acceso al experimento real
- Experimentos: acceso a ficheros con resultados
- Ayuda: información de la práctica

Acceso Laboratorio Remoto: Sistemas con Circuitos

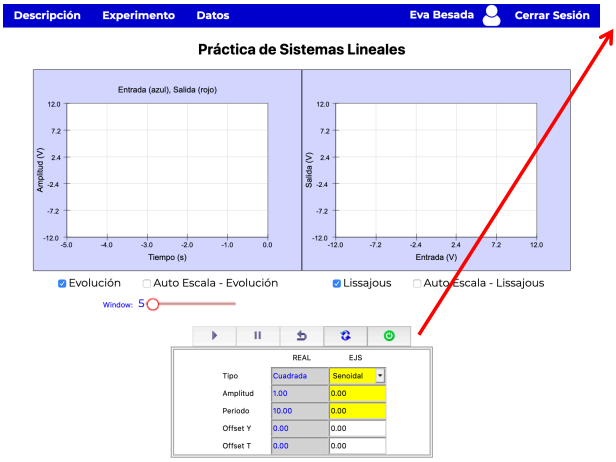


Tras pinchar el icono PRACTICA (Experimento en el menú superior) se accede a la página que permite parametrizar la entrada y observar la respuesta del circuito

Aparece un mensaje en el que se indica, según el caso:

a) Si la conexión se ha realizado con éxito y el tiempo de duración máximo de la conexión. Pulsar aceptar para poder hacer la experiencia
b) Si la conexión no se ha realizado con éxito porque hay otra persona conectada en ese momento. En este caso, habrá que esperar a que se desconecte

Acceso Laboratorio Remoto: Sistemas con Circuitos



El menú de botones intermedio, sirve para interactuar con el circuito

- Lanzar el experimento
- Parar el experimento (aplicarle tensión nula)
- Reiniciar el experimento (volver al punto iniciar)
- Modificar parámetros de señal de entrada
- Conectar/desconectar con el circuito

Tras entrar en la página,

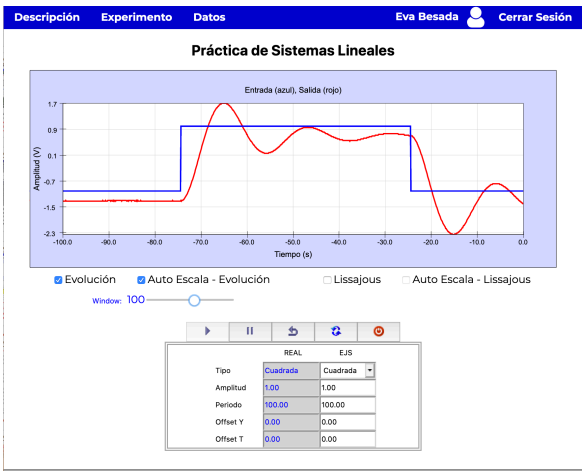
1. Pulsar botón de inicio
2. En cualquier momento, modificar valores de la entrada en columna (EJS) y pulsar para trasladarlos al sistema (REAL)

Acceso Laboratorio Remoto: Sistemas con Circuitos

Resultados experimentales

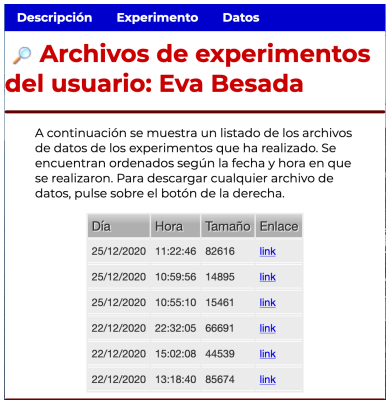
Se muestran en dos gráficas (Evolución entra x(t) y salida y(t) frente al tiempo, Lissajous (y(t) vs x(t)):

- *Activarlas/desativarlas de forma independiente con su selector*
- *Elegir si se desea que se reescalen los ejes a los valores de las señales*
- *Limpiar las gráficas clicando dos veces seguidas sobre cualquiera de ellas*



- *El deslizador window permite ajustar el valor de la ventana temporal de la grafica de evolución*
- *Los resultados se almacenan automáticamente en un fichero dDatos*
- *el servidor de la práctica. Para acceder a él, elegir en el menú superior*

Acceso Laboratorio Remoto: Sistemas con Circuitos



Día	Hora	Tamaño	Enlace
25/12/2020	11:22:46	82616	link
25/12/2020	10:59:56	14895	link
25/12/2020	10:55:10	15461	link
22/12/2020	22:32:05	66691	link
22/12/2020	15:02:08	44539	link
22/12/2020	13:18:40	85674	link

Tras pinchar el menu DATOS se observan todos los ficheros que se han almacenado en el servidor para cada alumno, ordenados por fecha y hora.

El alumno puede descargar los ficheros en su PC, tablet o movil pulsando el enlace correspondiente.

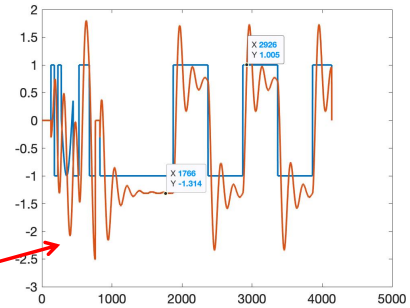
Es un fichero de texto, que se puede cargar desde Matlab con la orden load nombrefichero.txt

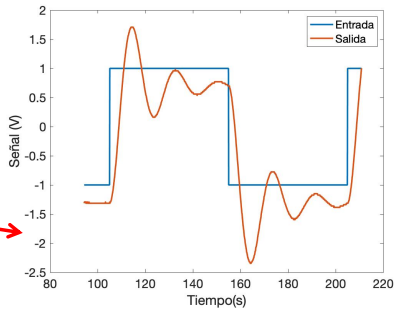
La orden creará una variable (matriz) que se llame igual que el nombre del fichero (nombrefichero), y que en cada fila tienes los datos de un instante t y que en tiene la siguiente información en las columnas primera (tiempo), segunda (entrada o referencia), cuarta (salida). El resto de las columnas no se usan en las practicas de Sistemas Lineales.

Acceso Laboratorio Remoto: Sistemas con Circuitos

Como los datos del fichero recogen toda la información del experimento se recomienda un código similar al siguiente para recortar y representar el trozo deseado

```
data=load('eva_20201225_112246.txt'); %Guardo la matriz en data
%Representar la entrada y salida directamente (no frente al tiempo)
%Esto hace que en el eje x se muestre el indice de cada valor
figure
plot(data(:,2));hold on; %Entrada (frente al indice)
plot(data(:,4));         %Salida  (frente al indice)

%Tras mirar la grafica, elegir el tramo que se desea
index=1766:2926;
%A continuación pintar la grafica con el tramo elegido y los instantes de
%tiempo en el eje x
figure
plot(data(index,1),data(index,2));hold on; %Entrada (frente al tiempo)
plot(data(index,1),data(index,4));         %Salida  (frente al tiempo)
xlabel('Tiempo(s)');ylabel('Señal (V)')
legend('Entrada','Salida');
```





Apéndice G

Practica de Sistemas Lineales: Identificación de un sistema mediante su diagrama de Bode experimental

Esta práctica se lleva realizando, de forma presencial, en el laboratorio de Sistemas Lineales de diferentes titulaciones desde el curso 2007/2008.

Sin embargo, durante el curso 2020-2021, debido a la situación semi-presencial de los laboratorios de la asignatura de Sistemas Lineales de la titulación de Ingeniería Electrónica en Comunicaciones y a la disponibilidad de las herramientas desarrolladas en este PIMCD, ha sido totalmente realizada de forma remota.

A continuación, se muestra el guión de la práctica que se les proporciona a los alumnos, actualizado con las modificaciones necesarias para reflejar la forma en la que han realizado la práctica durante este curso académico.

G.1. Guión de la práctica

A continuación se presentan los objetivos, la descripción y las tareas en las que se estructura la práctica de identificación de la función de transferencia de un sistema real (en este caso, el circuito analógico reconfigurable) mediante la obtención y análisis de su diagrama de Bode experimental.

G.1.1. Objetivos

El objetivo de esta práctica es obtener la función de transferencia de un sistema continuo analizando su respuesta en frecuencia. Para alcanzarlo, se obtendrá el diagrama de Bode experimental, proporcionando a la entrada del sistema una señal sinusoidal y observando en la respuesta del mismo el desfase con la señal de entrada y el cambio de la amplitud. Con los datos recogidos experimentalmente, se construirá el diagrama de Bode correspondiente. A continuación, este diagrama será analizado para obtener la función de transferencia cuyo diagrama de Bode se ajuste mejor al diagrama de Bode obtenido experimentalmente.

G.1.2. Descripción de la práctica

El sistema que se desea identificar es el que se encuadra en fuxia en la siguiente figura, y esta formado por dos polos simples en serie y, opcionalmente (según la disposición de los jumpers representados con cuadrados amarillos) un lazo cerrado entre la entrada y salida de los dos elementos anteriores. Por lo tanto, se puede comportar como un sistema sobreamortiguado o subamortiguado.

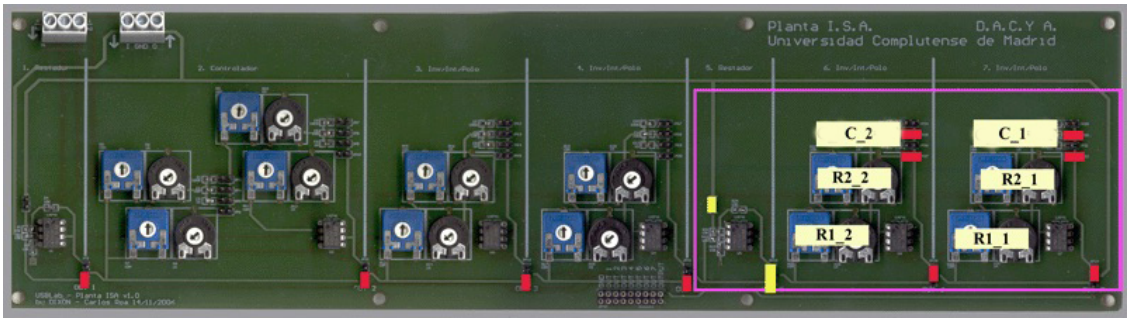


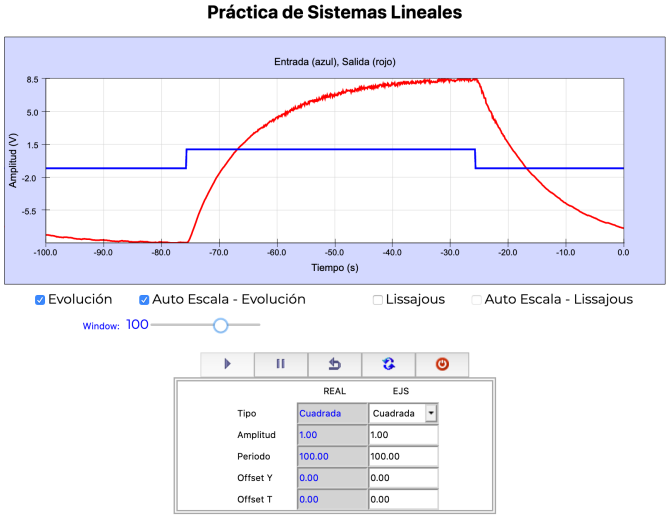
Figura G.1: Configuración del circuito para la práctica del diagrama de Bode

Para poder excitar el circuito de forma remota se utilizarán las herramientas software y hardware que se describen en la sección G.2.3.

G.1.3. Realización de la práctica

Las tareas que debe realizar el alumno a lo largo de la práctica son:

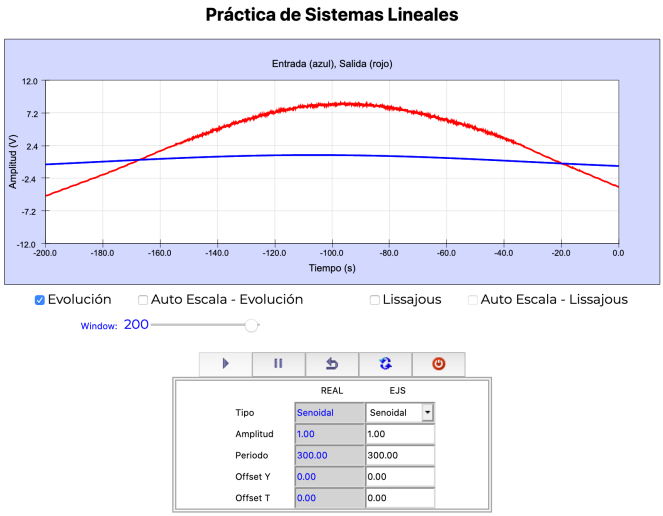
1. Observar la respuesta del sistema ante una onda cuadrada de duración adecuada. Para ello, el alumno aplicará sobre el circuito una onda cuadrada de amplitud 1 y duración adecuada para que la respuesta alcance el estacionario. En la siguiente figura se muestra como configurar el entorno gráfico de la experiencia remota para esta parte de la práctica y para una configuración determinada de la planta. El alumno deberá elegir el periodo de la onda cuadrada para conseguir observar el estacionario de la respuesta de forma adecuada.



El análisis de esta respuesta permitirá determinar dos cosas:

- Si existe un bias (desviación) en la respuesta, y en caso afirmativo su valor.
- Las características del estacionario y transitorio de la planta ante la entrada escalón escalón, y en particular, el tiempo de asentamiento del sistema.

2. La respuesta del sistema ante sinusoides de diferentes frecuencias y amplitudes. En la siguiente imagen se observa como configurar el entorno gráfico en este caso, para una frecuencia y amplitud determinada.



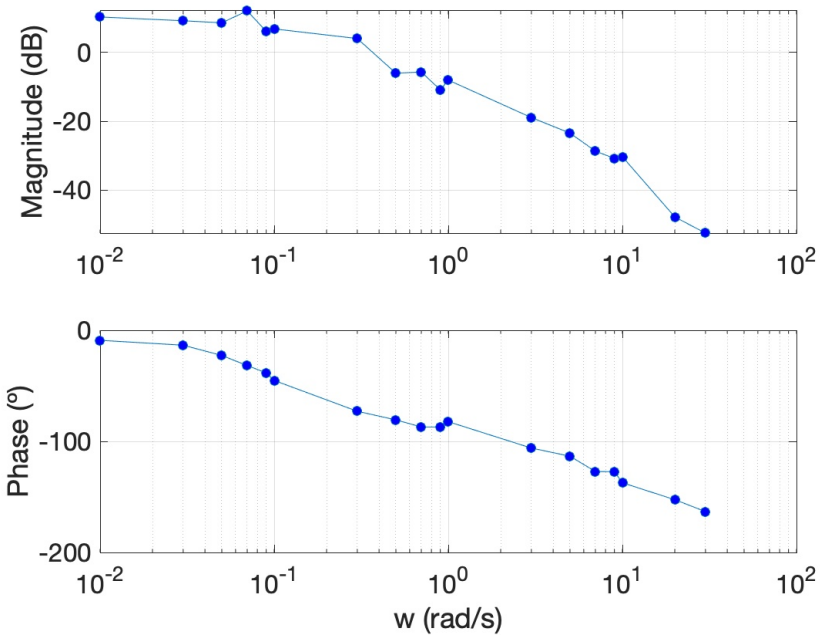
El rango de frecuencias que se debe utilizar es $w = [0{,}01, 10]$ rad/s, equiespaciadas logaritmicamente. Es conveniente empezar con una sinusoide de amplitud 1, y según se vaya atenuando la señal de salida, duplicar la amplitud de la sinusoide de entrada (teniendo en cuenta que la amplitud máxima permitida es de 12 voltios). Para evitar tomar datos del transitorio, después de cada cambio de frecuencia y/o amplitud, se debe esperar el tiempo de asentamiento determinado en el paso anterior para analizar únicamente la respuesta permanente a la sinusoide. A lo largo de este proceso el alumno debe crear una tabla, en la que almacene:

- La frecuencia de la sinusoide aplicada.
- La amplitud de la señal de entrada.
- La amplitud de la señal de salida.
- El desfase entre la señal de entrada y la señal de salida.

Estos valores pueden irse calculando sobre los resultados que se observan en la página web que da acceso al laboratorio remoto, aunque es preferible hacerlo, al ser el método más preciso y permitir que se corrija el bías, a través de la representación en Matlab de los datos de la experiencia que han sido almacenados en la página web del laboratorio. Además, para obtener una caracterización precisa del desfase introducido por la planta, es conveniente utilizar las figuras de Lissajous y seguir el procedimiento que se explica en la sección G.2.2.

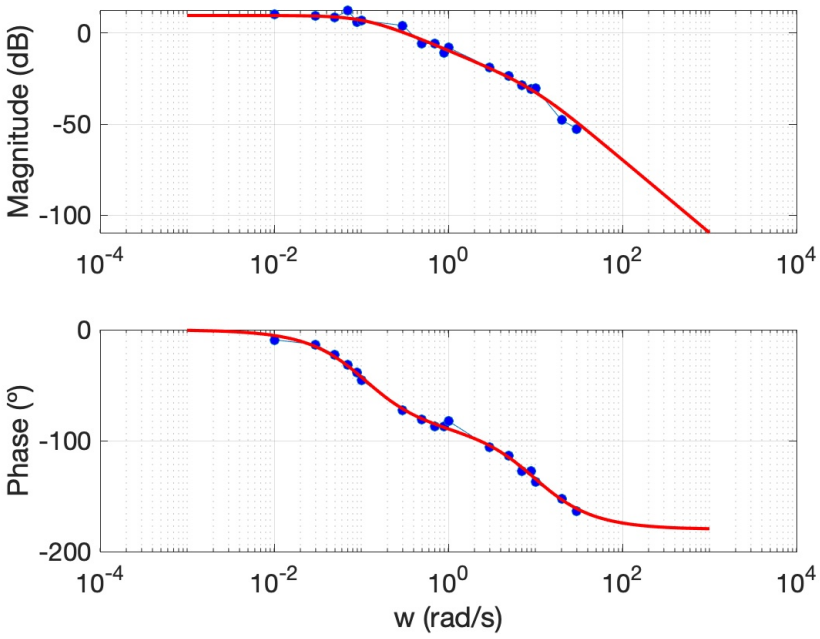
3. Con la tabla de valores obtenidos, se representará el diagrama de Bode experimental y se analizará dicho diagrama para identificar la función de transferencia del sistema. Hay que tener en cuenta que los datos experimentales pueden tener error, por lo que hay que intentar ajustar el diagrama de forma genérica, en promedio, y no a cada valor particular. Además, es recomendable que el alumno lea la sección G.2.1 para entender el tipo de función de transferencia que debe esperar del sistema configurado en el circuito.

   En la siguiente figura se representa, a modo de ejemplo, el diagrama de Bode que se ha obtenido con los datos experimentales de una configuración concreta de la planta.



4. Se superpondrá gráficamente el diagrama de Bode obtenido experimentalmente con el diagrama de Bode teórico del sistema identificado con el fin de poder compararlos y ver si coinciden. De no ser así, el alumno deberá modificar la ganancia y los ceros y/o polos del sistema hasta que el diagrama teórico se ajuste al experimental.

   En la siguiente figura se muestra un ejemplo de la superposición de ambos diagramas.



5. Finalmente, aplicará a la función de transferencia obtenida una onda cuadrada con las mismas características que la del primer paso, y comprobará si el modelo obtenido y el sistema físico tienen la misma respuesta temporal (para ello, se recomienda representar los resultados de ambos casos sobre la misma gráfica).

G.2. Información adicional sobre la práctica

A continuación presentamos diferente información de utilidad para profundizar en diferentes aspectos y formas de proceder a lo largo de esta práctica.

G.2.1. Estudio teórico del circuito

El circuito de la planta, que se encuadra en fuxia en Figura G.1, esta formado por dos polos simples colocados en serie, y opcionalmente un restador que creará un lazo cerrado entre la entrada y la salida del sistema. En el esquema, los rectángulos en rojo representan la posición de jumpers que se mantendrán en cualquier caso, y la disposición de los amarillos, de elementos opcionales o que se pueden colocar, según la configuración elegida, en dos posiciones diferentes. Información adicional del circuito completo se encuentra disponible en la página web que da acceso a la práctica.

Para analizar el comportamiento del circuito, estudiaremos el comportamiento de cada componente por separado. Cada uno de las componentes Inv/Int/Polo que está en el interior del recuadro fuxia se encuentra conectado en la configuración de un polo simple, y por lo tanto, la relación entre la entrada y salida de cada uno de estos subsistemas viene data por la siguiente función de transferencia:

$$G(s) = -\frac{R_{2\_}}{R_{1\_}}\frac{1}{R_{2\_}C_{\_}s+1}$$

En el caso en el que únicamente los dos polos se encuentran conectados en serie (y el restador no forme parte del circuito) la función de transferencia del sistema es la siguiente, que se obtiene como el producto de las funciones de transferencia dos polos simples en serie (y por lo tanto el comportamiento del sistema será el de un segundo orden sobreamortiguado).

$$G(s) = \frac{R_{2\_1}R_{2\_2}}{R_{1\_1}R_{1\_1}R_{1\_2}}\frac{1}{(R_{2\_1}C_{\_1}s+1)(R_{2\_2}C_{\_2}s+1)}$$

En el caso en el que se cierre el lazo existente entre la entrada y la salida, utilizando el elemento restador, la función de transferencia del sistema que se va a analizar es la siguiente, que tiene en el denominador un polinomio de segundo orden en‘s’ con polos complejos conjugados, y por lo tanto el comportamiento del sistema será el de un segundo orden subamortiguado.

$$G(s) = \frac{R_{2\_1}R_{2\_2}}{R_{1\_1}R_{1\_2}(R_{2\_1}C_{\_1}s+1)(R_{2\_2}C_{\_2}s+1)+R_{2\_1}R_{2\_2}}$$

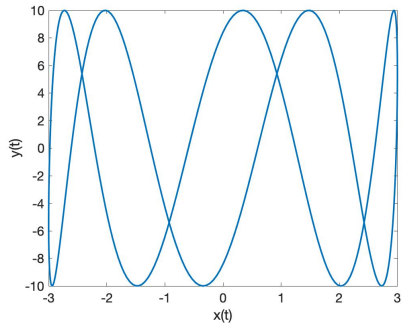
G.2.2. Figuras de Lissajous

Las curvas de Lissajous se obtienen cuando se representan gráficamente una frente a otra (una en el eje x y la otra en el eje y) dos señales de la forma:

$$x(t) = A_x sin(w_x t + \phi_x)$$

$$y(t) = A_y sin(w_y t + \phi_y)$$

Se dice que las dos ecuaciones anteriores constituyen la representación paramétrica de dichas curvas. Variando los valores de $A_x, A_y, w_x, w_y, \phi_x$ y ϕ_y se obtienen las distintas curvas de la familia. Por ejemplo, para $A_x = 3, A_y = 10, w_x = 2$ rad/s,$w_y = 10$rad/s, $\phi_x = 0$ rad y $\phi_y = 1$ rad se obtiene la siguiente curva, representando con la orden plot la $x(t)$ frente a la $y(t)$.



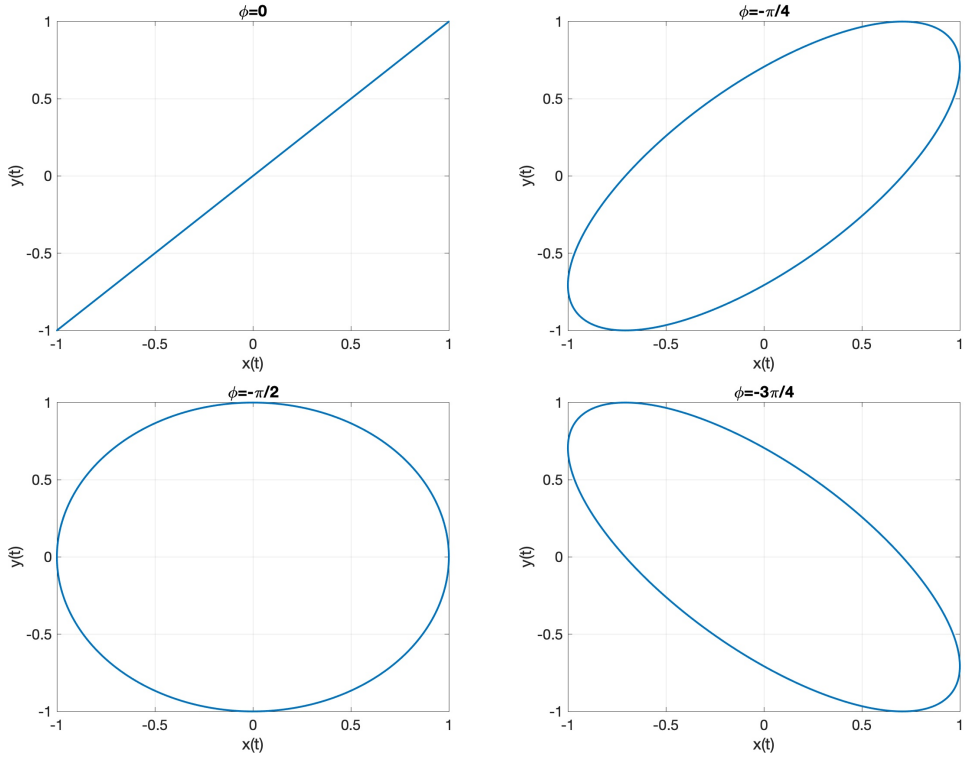
En nuestro caso, podemos considerar que la señal de entrada a la planta ($x(t)$) y la señal salida ($y(t)$) en régimen permanente son de la forma:

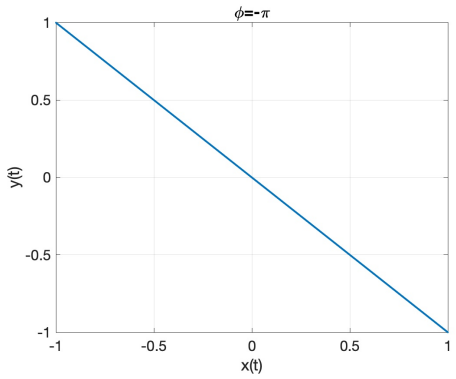
$$x(t) = A_x sin(wt)$$

$$y(t) = A_y sin(wt + \phi)$$

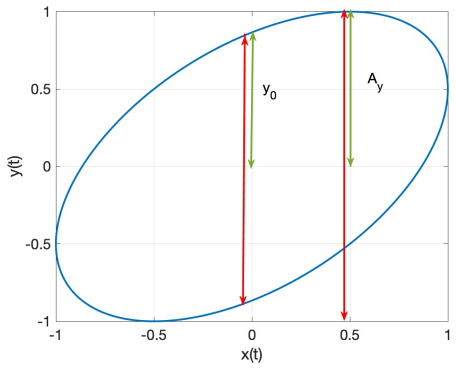
ya que la frecuencia de la señal de salida es igual a la de la señal de entrada y lo que nos interesa calcular es el desfase relativo entre las dos señales.

En el siguiente grupo de figura se muestran las curvas que se obtienen a una frecuencia w deteterminada, con amplitudes iguales y diferentes desfases.





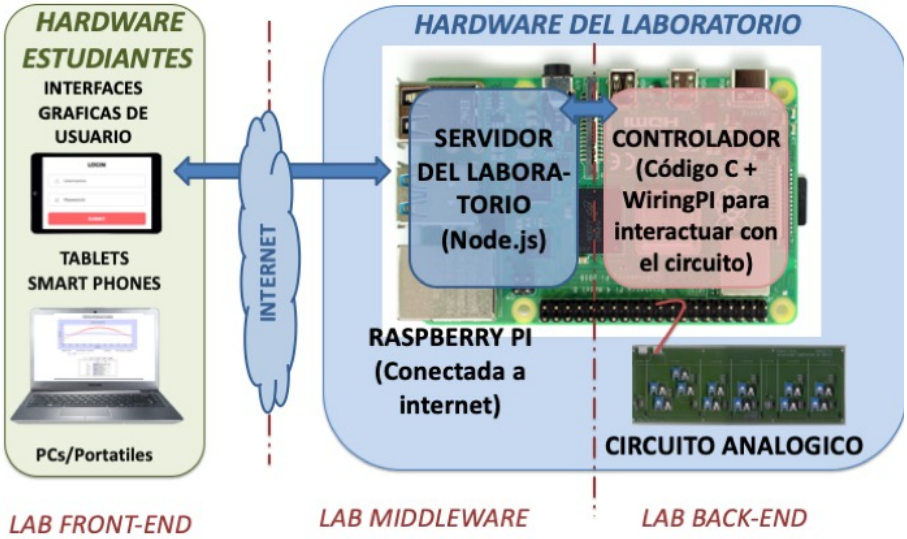
El desfase es fácilmente calculable tomando los cortes de la figura con el eje x (y=0) o con el eje y (x=0). La siguiente figura muestra cómo calcular el desfase relativo con el corte de la gráfica con el eje y, y la fórmula $\phi = arcsin(y_0/A_y)$. En el caso de descargarse los valores de internet, una vez recortado el tramo adecuado y desechados los valores del transitorio, para obtener y_0 (marcado en la linea verde) es suficiente con buscar los puntos de las x cuyo valor es casi cero y observar el valor correspondiente de la y. En el caso de hacerlo directamente sobre la gráfica de la página web, se recomienda tomar los valores pico a pico (no respecto a cero, si no los valores extremos, marcados en rojo en la gráfica), para evitar los errores asociados al offset.



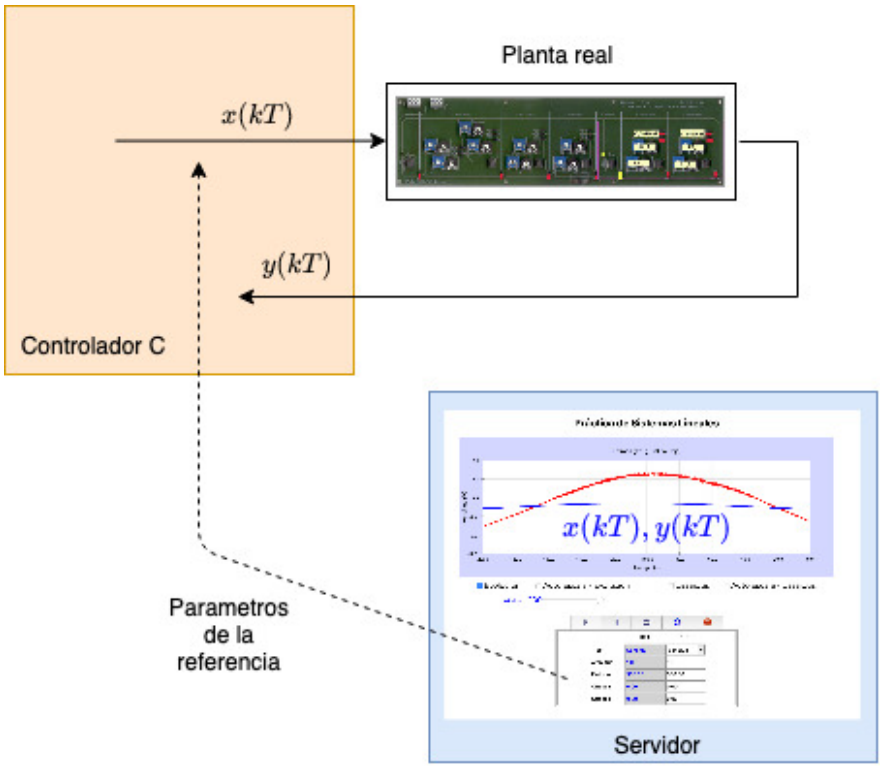
Finalmente, es importante tener en cuenta que la expresión para el desfase $\phi = arcsin(y_0/A_y)$ solo da valores en el intervalo [0,90] grados (ni da el signo ni cubre el rango [0,360]). Por lo tanto, a la hora de calcular el desfase final para el diagrama de Bode hay que comparar si la señal de entrada adelanta o retrasa a la de salida, y hacer las correcciones necesarias.

G.2.3. Obtención de los datos experimentales de forma remota

Para que los alumnos puedan estudiar de forma remota la respuesta del circuito ante distintas entradas, se ha conectado el circuito analógico a una Raspberry PI, sobre la que se ejecuta 1) un programa (denominado controlador de forma genérica) que calcula la señal de referencia, la aplica sobre la planta y mide su respuesta; y 2) un servidor web que permite que los alumnos accedan a los parámetros de la señal de entrada (referencia) y observen la evolución de diferentes señales desde sus casa. La figura G.2 esquematiza los elementos hadware y software involucrados en el proceso y la relación que se establece, desde el punto de vista de Sistemas lineales entre las señales que se muestran a través de la página web y las señales de la planta (generadas/medidas por el código de Control).



(a) Relación entre los elementos hardware y software del laboratorio



(b) Relación entre los elementos desde el punto de vista de Sistemas Lineales

Figura G.2: Relación entre los elementos del laboratorio remoto

Apéndice H

Trabajo de Evaluación Continua de Procesamiento de Señales: Identificación de un sistema a partir de su respuesta a señales aleatorias

En la asignatura de Procesamiento de Señales de la titulación de Ingeniería Electrónica en Comunicaciones, los alumnos realizan, desde el curso 2008-2009, una tarea en el que simulan la respuesta de un sistema discreto desconocido a una señal aleatoria ergódica (p.e. una gaussiana o una uniforme) y a través de las relaciones que se establecen entre los espectros de las autocorrelaciones de la señal de entrada y de salida, y el espectro de la correlación cruzada de ambas señales, identifican la función de transferencia del sistema.

Durante este PIMCD se ha desarrollado el software necesario para poder realizar esta experiencia sobre un sistema real (en concreto, el circuito reconfigurable utilizado en la asignatura de Sistemas Lineales) y se ha generado la documentación para el Trabajo de Evaluación Continua correspondiente. En este apéndice se muestra el guión que se les proporcionará a los alumnos para que realicen este nuevo trabajo a partir del próximo curso académico.

H.1. Guión del Trabajo de Evaluación Continua

A continuación se presentan los objetivos y las tareas de identificación de sistemas a partir de su respuesta a señales aleatorias que se deben realizar en esta trabajo de evaluación continua.

H.1.1. Objetivo

El objetivo de este trabajo de evaluación continua es obtener la función de transferencia asociada a los modelos de dos sistemas discretos a partir del estudio en frecuencia de su respuesta a señales aleatorias.

H.1.2. Metodología de análisis

Para alcanzar el objetivo anterior, el alumno realizará las siguientes tareas sobre cada uno de los dos sistemas:

1. Excitar y registrar el sistema con una señal ergódica.

2. Calcular la autocorrelación de la señal de entrada, la autocorrelación de la señal de salida, y la correlación cruzada entre la entrada y la salida.
3. Calcular la Transformada Rápida de Fourier (FFT) de las tres funciones obtenidas en el paso anterior.
4. Obtener el diagrama de Bode del sistema a partir de la razón entre las FFT anteriormente obtenidas.
5. Obtener la función de transferencia correspondiente al diagrama de Bode.
6. Comprobar de forma gráfica si el diagrama de Bode obtenido experimentalmente coincide con el diagrama de Bode de la función de transferencia estimada.

H.1.3. Sistema simulado

- La respuesta de un sistema discreto a una señal aleatoria ergódica se encuentra recogida en la variable matricial data almacenada en el fichero Respuesta1.mat. Para analizar dicha matriz, el alumno ejecutará la orden load Respuesta1.mat y obtendrá las señal de entrada y salida, respectivamente, a partir de la segunda y tercera columna de la matriz, mientra que los instantes de tiempo de cada una de las señales los obtendrá de la primera columna de la matriz.

  A continuación, el alumno deberá obtener la función de transferencia del sistema con el procedimiento descrito en la sección anterior y determinar el tipo de la señal de entrada ergódica que se le ha aplicado al sistema.

H.1.4. Circuitos reales

- La respuesta de un circuito real formado por amplificadores operacionales, resistencias y condensadores, se puede obtener conectándose a internet con un navegador a la dirección http://XXX.XX.XX.XX siguiendo el procedimiento detallado en la sección H.2. El alumno: 1) observará la respuesta a la entrada aleatoria que elija, 2) descargará el fichero generado durante el experimento y lo cargará en Matlab mediante la orden data=load(*NombreFichero*.txt), 3) obtendrá de la variable data la señal de entrada (segunda columna), la señal de salida (cuarta columna) y los instantes de tiempo de cada señal.

  A continuación, el alumno deberá obtener la función de transferencia del sistema con el procedimiento descrito anteriormente.

Para realizar estas últimas tarea el usuario y password de cada alumno es el siguiente:

- Usuario: El mismo que tenéis en vuestra dirección de correo de la UCM. Es decir, si vuestro correo es abcd@ucm.es, el usuario en la página web del laboratorio es abcd.
- El password está formado por tres partes, separadas por un -:

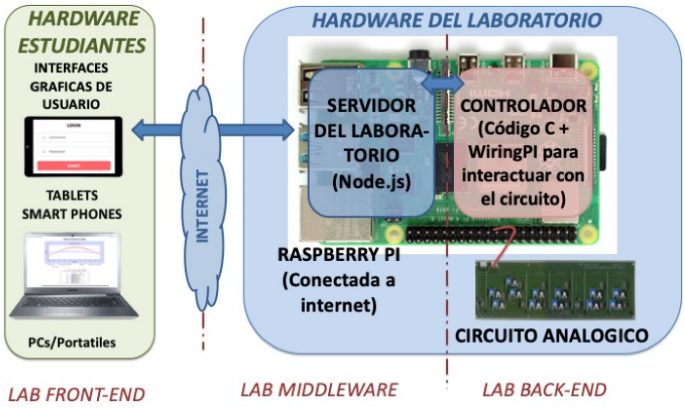
  1) las tres primeras letras del nombre del alumno (en mayúsculas), 2) los tres primeros número del teléfono consignado en la UCM, y 3) las tres últimas letras del NIF.

H.2. Información adicional del Trabajo de Evaluación Continua

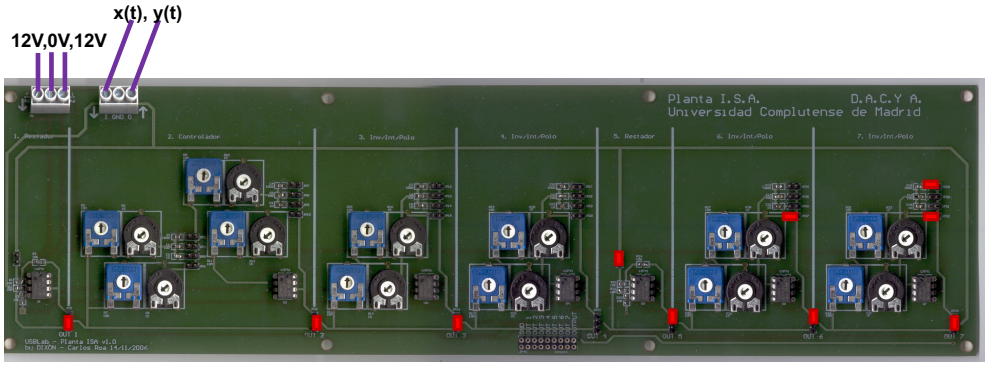
A los alumnos también se les proporciona, de forma adicional, unas transparencias (que se presentan a continuación) para que aprendan a utilizar la página web del laboratorio remoto.

Laboratorio Remoto de Sistemas con Circuitos

El laboratorio remoto de circuitos permite a los alumnos acceder desde su casa a los circuitos reconfigurables del laboratorio e interactuar con ellos (eligiendo que señales se aplican sobre el y observando su respuesta por pantalla y a través de ficheros)
Los elementos que dan soporte al laboratorio se muestran en el siguiente esquema, transparente para los alumnos, que acceden a el a través de una página web tal y como se explicará en las siguientes transparencias



Laboratorio Remoto de Sistemas con Circuitos



Circuito reconfigurable analógico

Es un circuito formado por amplificadores operacionales (741), potenciómetros, condensadores y jumpers (rojo), que según como se configue puede comportarse como un polo simple, doble, complejo conjugado, triple, etc.
Información adicional del circuito, puede verse en la página de ayuda (descripción) del laboratorio remoto (accesible tal y como se muestra a continuación).
En la asignatura de Procesamiento de Señales, el alumno utilizará el circuito con la configuración que seleccionen los profesores.

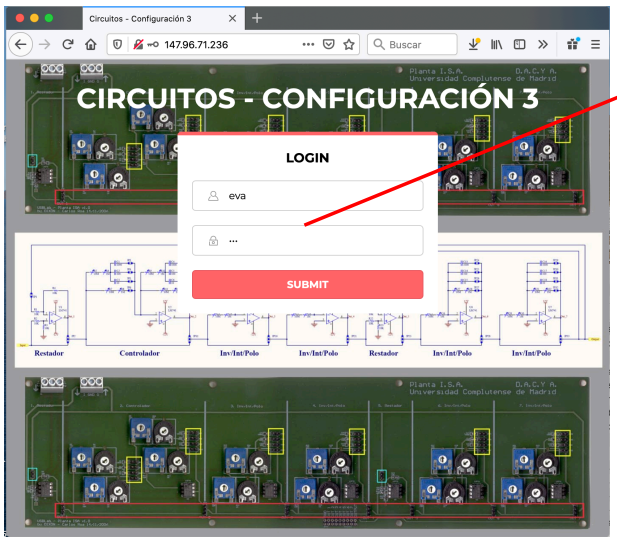
Acceso Laboratorio Remoto: Sistemas con Circuitos

Acceder a la página web de las practicas remotas, escribiendo en la URL, la IP indicada en cada experimento.

Ejemplo:
147.96.71.236

Se puede acceder a las prácticas remotas desde diferentes navegadores de tablets, móviles y PCs. Sin embargo, es preferible hacerlo desde un PC y Firefox.

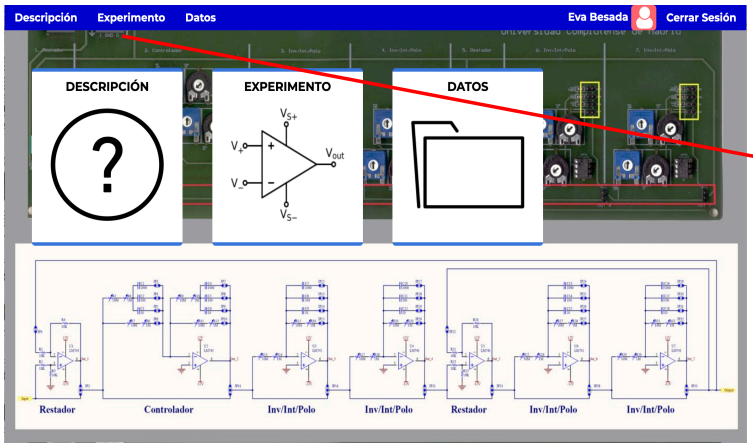
Acceso Laboratorio Remoto: Sistemas con Circuitos



Acto seguido, hay que introducir el usuario y la password del alumno

El alumno utilizará el usuario y password que se le indica en el guión de la práctica. Debe ser utilizado para todas las prácticas y TECs remotas que se realicen durante el curso 2020/21

Acceso Laboratorio Remoto: Sistemas con Circuitos



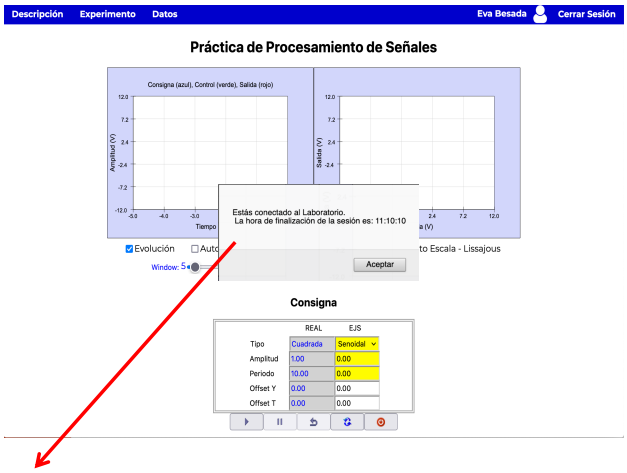
Terminar la sesión

Menú para navegar entre las opciones desde las distintas páginas web del laboratorio

A continuación, se muestra la página principal del laboratorio con iconos que enlazan a diferentes elementos:

- Descripción: información de la práctica
- Práctica: acceso al interfaz de la experiencia
- Datos: acceso a ficheros con resultados

Acceso Laboratorio Remoto: Sistemas con Circuitos



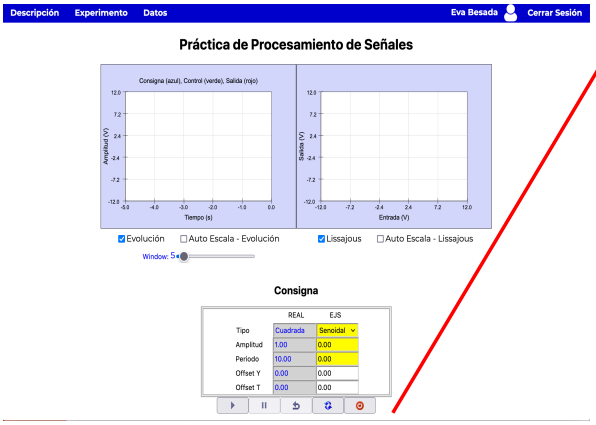
Tras pinchar el icono EXPERIMENTO se accede a la página que permite parametrizar la entrada y observar la respuesta del circuito

Aparece un mensaje en el que se indica, según el caso:

a) Si la conexión se ha realizado con éxito y el tiempo de duración máximo de la conexión. Pulsar aceptar para poder hacer la experiencia

b) Si la conexión no se ha realizado con éxito porque hay otra persona conectada en ese momento. En este caso, habrá que esperar a que se desconecte

Acceso Laboratorio Remoto: Sistemas con Circuitos



El menú de botones inferior, sirve para interactuar con el circuito

- Lanzar el experimento
- Parar el experimento (aplicarle tensión nula)
- Reiniciar el experimento (volver al punto iniciar)
- Modificar parámetros de señal de entrada
- Conectar/desconectar con el circuito

Tras entrar en la página,

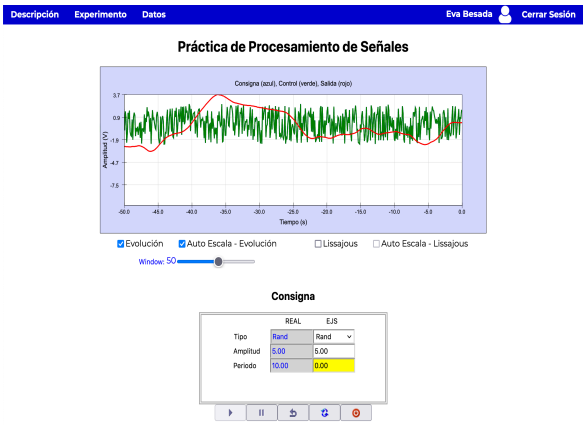
1. Pulsar botón de inicio
2. En cualquier momento, modificar valores de la entrada en columna (EJS) y pulsar para trasladarlos al sistema (REAL)

Acceso Laboratorio Remoto: Sistemas con Circuitos

Resultados experimentales

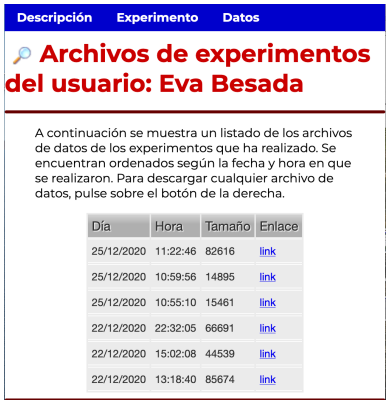
Se muestran en dos gráficas (Evolución entra x(t) y salida y(t) frente al tiempo, Lissajous (y(t) vs x(t)):

- *Activarlas/desativarlas de forma independiente con su selector*
- *Elegir si se desea que se reescalen los ejes a los valores de las señales*



- *Limpiar las gráficas clicando dos veces seguidas sobre cualquiera de ellas*
- *El deslizador window permite ajustar el valor de la ventana temporal de la grafica de evolución*
- *Los resultados se almacenan automáticamente en un fichero del servidor de la práctica. Para acceder a él, elegir en el menú superior la opcion Datos*

Acceso Laboratorio Remoto: Sistemas con Circuitos



Descripción Experimento Datos

Archivos de experimentos del usuario: Eva Besada

A continuación se muestra un listado de los archivos de datos de los experimentos que ha realizado. Se encuentran ordenados según la fecha y hora en que se realizaron. Para descargar cualquier archivo de datos, pulse sobre el botón de la derecha.

Día	Hora	Tamaño	Enlace
25/12/2020	11:22:46	82616	link
25/12/2020	10:59:56	14895	link
25/12/2020	10:55:10	15461	link
22/12/2020	22:32:05	66691	link
22/12/2020	15:02:08	44539	link
22/12/2020	13:18:40	85674	link

Tras pinchar el menu DATOS se observan todos los ficheros que se han almacenado en el servidor para cada alumno, ordenados por fecha y hora.

El alumno puede descargar los ficheros en su PC, tablet o movil pulsando el enlace correspondiente.

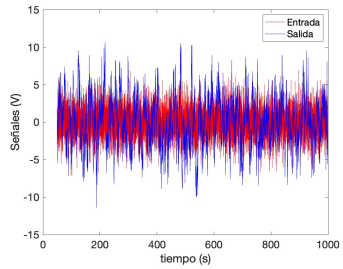
Es un fichero de texto, que se puede cargar desde Matlab con la orden load nombrefichero.txt

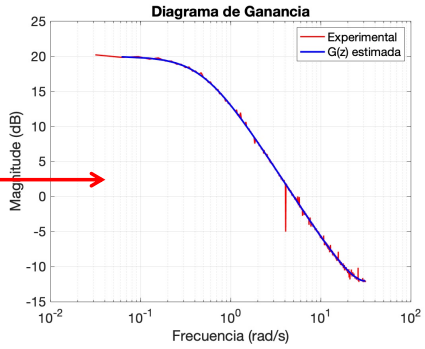
La orden creará una variable (matriz) que se llame igual que el nombre del fichero (nombrefichero), y que en cada fila tienes los datos de un instante t y que en las asignaturas de Procesamiento de Señales tiene la siguiente información en cada columna: primera (tiempo), segunda (entrada) y tercera (salida).

Acceso Laboratorio Remoto: Sistemas con Circuitos

Como los datos del fichero recogen toda la información del experimento, se recomienda un código similar al siguiente para recortar y representar el trozo deseado, y para obtener y analizar el espectro en frecuencia

```
close all;clear all;

%Cargamos fichero
datos=load('eva_XXXXXXXXXXX.txt');
%Elegimos parte de la señal
t=data(500:end,1);x=data(500:end,2);y=data(500:end,3);

%Representamos señales
figure
plot(t,x,'r',t,y,'b');
legend('Entrada','Salida');
xlabel('tiempo (s)'),ylabel('Señales (V)')

%Autocorrelaciones
[xc,lags] = xcorr(x,1000,'unbias');
[yc,lags] = xcorr(y,1000,'unbias');
% Potencia espectral de las señales:
Xc = fft(xc);Yc = fft(yc);
%Relacion espectros de potencia de las dos señales
Gain = sqrt(Yc./Xc);
N = length(Xc);half=round(N/2);
wc = 2*pi/N*[0:N-1]/Ts; %wc correspondiente
figure
%Solo dibujamos la mitad del espectro
wp=wc(1:half);Gainp=abs(Gain(1:half));
semilogx(wp,20*log10(Gainp),'r','Linewidth',1.5);

%Funcion de transferencia estimada
gs=tf(5,[1,0.5]);gz=c2d(gs,Ts);
[numz,denz]=tfdata(gz,'v')
[Hw,wd] = freqz(numz,denz);
wc=wd/Ts;
hold on,semilogx(wc,20*log10(abs(Hw)),'b','Linewidth',2);
title('Diagrama de Ganancia');
xlabel('Frecuencia (rad/s)'); ylabel('Magnitude (dB)');
legend('Experimental','G(z) estimada'),grid on
```





Apéndice I

Práctica de Control con una Red de Adelanto

Esta práctica se lleva realizando, de forma presencial, en el laboratorio de Control de Sistemas de diferentes titulaciones desde el curso 2007/2008.

Sin embargo, durante el curso 2020-2021, debido a la situación semi-presencial de los laboratorios de la asignatura de Control de Sistemas de la titulación de Ingeniería Electrónica en Comunicaciones y a la disponibilidad de las herramientas desarrolladas en este PIMCD, ha sido mayoritariamente realizada de forma remota, ya que los alumnos han diseñado la red de adelanto y probado su versión discreta desde sus casas, y asistido únicamente a una breve sesión del laboratorio presencial para probar el funcionamiento del controlador continuo.

En las siguientes secciones, se muestra la documentación de la práctica que se les proporciona a los alumnos, actualizada con las modificaciones necesarias para reflejar la forma en la que han realizado la práctica durante este curso académico.

I.1. Guión de la práctica

A continuación se presentan los objetivos, la descripción y las tareas en las que se estructura la práctica de diseño e implementación de una red de adelanto, aplicada sobre el circuito analógico reconfigurable.

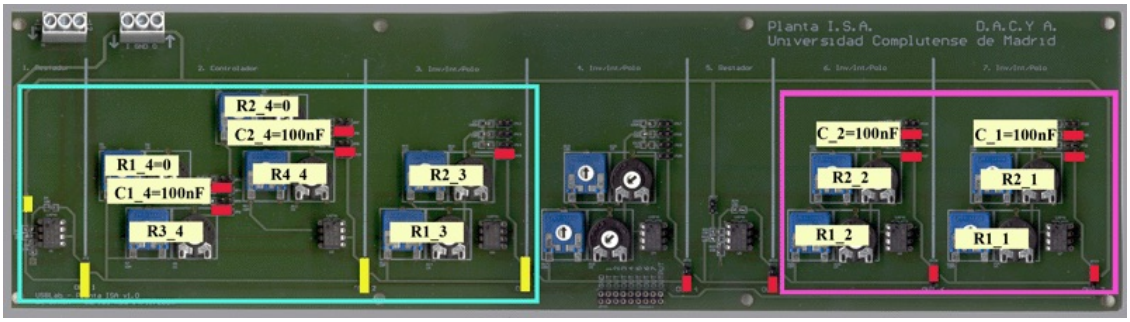
I.1.1. Objetivos

Los objetivos de esta práctica son:

- Diseño de un controlador continuo para una planta real, utilizando una red de adelanto con el fin de que el sistema realimentado verifique unas especificaciones dadas para el error estacionario y el margen de fase.
- Caracterización del funcionamiento del controlador continuo diseñado sobre una versión simulada de la planta.
- Discretización del controlador continuo y su implementación sobre la planta utilizando un proceso de control implementado en una Raspberry PI.
- Implementación del controlador continuo mediante un circuito analógico.

I.1.2. Descripción de la práctica

La planta que se desea controlar, utilizando una red de adelanto, es la encuadrada en la parte fuxia de la siguiente figura. Las características específicas de dicho sistema se encuentran recogidas en la sección I.2.1. Los elementos enmarcados en azul constituyen una red de adelanto continua, que se puede incluir o eliminar del circuito modificando la posición de los jumpers representados en amarillo de la parte inferior del circuito.



El primer objetivo de la práctica es diseñar un controlador mediante una red de adelanto, de forma que el sistema en lazo cerrado cumpla unas especificaciones dadas para el error y el margen de fase. El controlador diseñado debe incluir también el factor por el que es necesario multiplicar la ganancia de la planta para que el sistema (planta más compensador) tenga la ganancia necesaria para mejorar el error estacionario ya que no se puede modificar la ganancia de la planta.

Una vez calculadas los parámetros que caracterizan el comportamiento de la red de adelanto (incluyendo entre ellos el factor de ganancia), se:

1. Analizará el funcionamiento del sistema en lazo cerrado (formado por la planta + red + el factor de ganancia y una realimentación unitaria) mediante simulación. Para llevar a cabo el análisis se utilizará Simulink y se compararán:

   a) El comportamiento del lazo cerrado con todos los elementos (la planta + red + el factor de ganancia y la realimentación unitaria) en continuo.

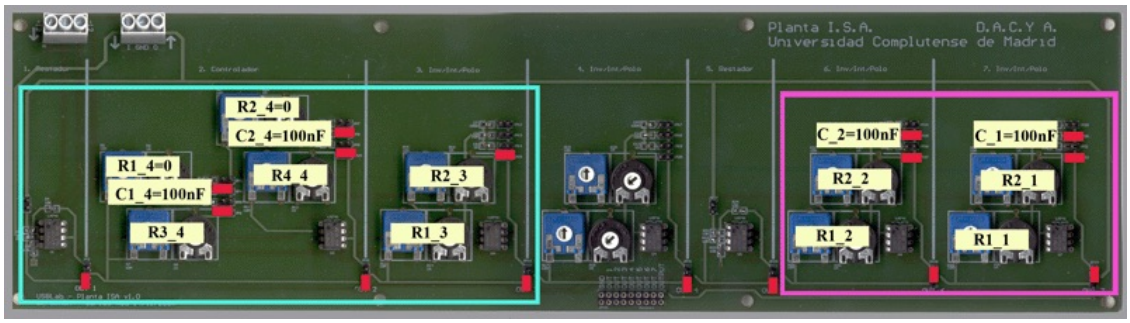
   b) El comportamiento del lazo cerrado anterior más las saturaciones ($\pm 12V$) que hay en la entrada y salida de la planta.

   c) El comportamiento del lazo cerrado anterior con la planta y el controlador discretizado con el periodo que se indique al inicio de la sesión de prácticas.

2. Se controlará la planta real a partir de la implementación de controlador + ganancia en discreto mediante una ecuación en diferencias implementada en un programa de control que se está ejecutando en una Raspberry PI, a cuyos parámetros se pueden acceder remotamente, desde el PC, portatil o tablet del alumno mediante una página web. En la sección I.2.2 se describe el procedimiento mediante el cual los alumnos discretizarán el controlador y modificarán sus parámetros a través de internet.

   Para poder controlar el sistema desde la Raspberry PI, es necesario que los jumpers representados en amarillo en la parte inferior del circuito anterior se desplacen a su situación inferior, y que se elimine el jumper que se encuentra antes del restador. Esto es debido a que los cálculos del controlador discreto y el cierre del lazo de control se realizan en el programa de controlador que se ejecuta en la Raspberry PI, y por lo tanto, el módulo de Control del circuito análogico reconfigurable y el sumador/restador que cierra el lazo deben estar deshabilitados.

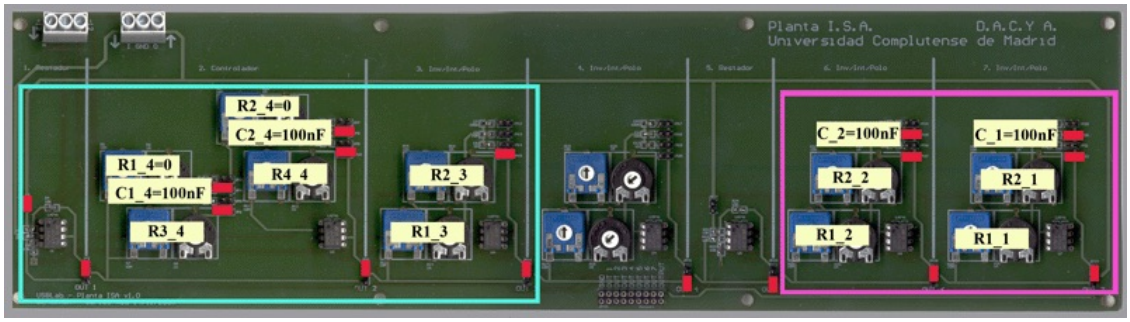
La configuración correcta, que será en la que se encuentra la planta durante el periodo en el que los alumnos deben comprobar los controladores discretos a través de internet, es la que se muestra en la siguiente figura.



3. Se controlará la planta real a partir de la implementación de controlador + ganancia en continuo mediante la configuración del controlador del circuito analógico reconfigurable (enmarcado en azul en la siguiente figura) y el sumador/restador que cierra el lazo de control.

   Para poder implementar dicho controlador es necesario calcular el valor de las resistencias que hacen que dicho circuito cumpla las especificaciones del controlador diseñado, y que serán sintonizadas por los alumnos durante la sesión presencial del laboratorio. En este caso, utilizaremos la Raspberry PI para aplicar la referencia y ver la respuesta del lazo cerrado.

   El controlador y el lazo cerrado se incluyen en el circuito colocando en la parte superior los jumpers amarillos de la parte inferior del primer circuito, e incluyendo el jumper del restador. La configuración correcta se presenta en la siguiente figura.



   El procemiento a seguir para calcular los valores de los elementos del circuito que intervienen en la red y para realizar esta parte de la práctica, de forma presencial en el laboratorio, se describe en la sección I.2.3.

I.1.3. Realización de la práctica

La práctica se puede estructurar en las siguientes etapas:

1. Diseñar mediante Matlab una red de adelanto de forma que el sistema compensado verifique unas especificaciones dadas para el error estacionario y el margen de fase. La función de transferencia de la planta, que se muestra a continuación, depende del valor elegido para la resistencias **R1_1**, **R2_1**, **R1_2** y **R2_2** y los condensadores **C_1** y **C_2**.

$$G_p(s) = \frac{R_{2\_1}R_{2\_2}}{R_{1\_1}R_{1\_2}} \frac{1}{(R_{2\_1}C_{\_1}s+1)(R_{2\_2}C_{\_2}s+1)}$$

   Al inicio de cada sesión de prácticas se indicará a los alumnos tanto las especificaciones para el error estacionario y el margen de fase como los valores de las resistencias elegidas para la planta.

2. Comprobar con Simulink el funcionamiento de la red sobre la función de transferencia de la planta y ver si se verifican las especificaciones dadas para la respuesta del sistema ante una entrada escalón. Observar las características de la respuesta del sistema realimentado (t_s, M_p, e_{ss}, t_r) a una entrada escalón.

3. Comprobar con Simulink el funcionamiento del lazo cerrado, tras añadir las saturaciones propias del circuito y discretizar el controlador con el periodo de muestreo que se indique al inicio de la sesión de prácticas. Comparar los resultados obtenidos con los del caso anterior, para determinar si la discretización y las saturaciones degradan significativamente la respuesta del lazo cerrado.

4. Comprobar el funcionamiento del controlador discretizado sobre la planta real. Para ello, seguir el proceso que se indica en la sección I.2.2.

5. Calcular, siguiendo el proceso que se explica en I.2.3, el valor de las resistencias para que el Controlador del circuito analógico se comporte como la red de adelanto activa diseñada en el primer apartado. Sintonizar los potenciómetros variables del controlador del laboratorio con los valores de las resistencias calculadas, e introducir los elementos relacionados con el controlador en el circuito. Observar la respuesta del sistema siguiendo el procedimiento que se indica en la sección I.2.3.

   Ante cualquier duda sobre las conexiones finales de los circuitos, el alumno debe acudir al profesor de prácticas o al técnico de laboratorio.

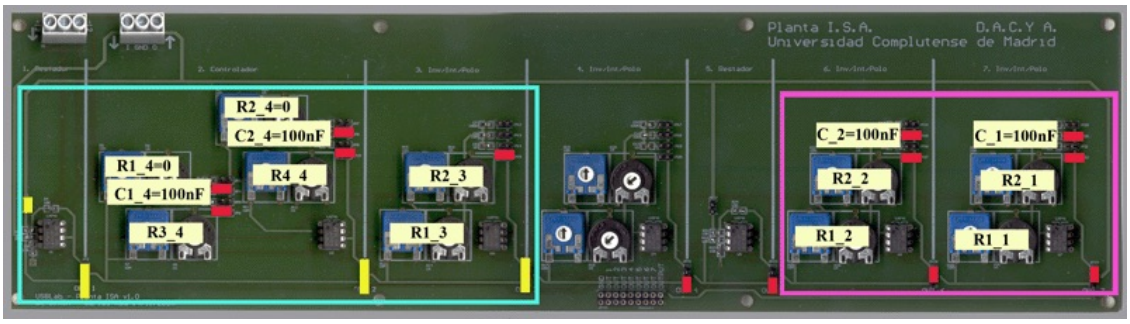
Finalmente, indicar que las etapas 1, 2, 3 y 4 pueden ser realizadas desde los equipos de los alumnos, sin que estos tengan que estar de forma presente en el laboratorio. Esto se debe a que para las tres primeras tareas el alumno utilizará Matlab y Simulink (herramientas de las que disponen los alumnos debido al acuerdo entre la UCM y Matlab) y que para la cuarta se ha dado acceso a la experiencia a través de internet. Además, el cálculo de las resistencias y controladores de la red también se puede realizar de forma remota, siendo únicamente necesario que los alumnos realicen la parte experimental de la tarea 5 de forma presencial, ya que durante esta fase deben ajustar/sintonizar los ponteciómetros del Controlador con los valores que han calculado previamente.

I.2. Información adicional sobre la práctica

A continuacion presentamos diferente informacion de utilidad para profundizar en diferentes aspectos y formas de proceder a lo largo de esta práctica.

I.2.1. Estudio teórico de la planta

El circuito de la planta, que se encuentra remarcado en fuxia en la siguiente figura, esta formado por dos polos simples colocados en serie.



Para analizar su comportamiento, estudiaremos el comportamiento de cada componente por separado. Cada una de las componentes Inv/Int/Polo que lo constituyen se encuentra conectado en la configuración de un polo simple, y por lo tanto, la relación entre la entrada y salida de cada uno de estos subsistemas viene data por la siguiente función de transferencia:

$$G(s) = \frac{R_{2\_X}}{R_{1\_X}} \frac{1}{(R_{2\_X}C_{\_X}s+1)}$$

Como ambos polos están conectados en serie, la función de transferencia de la planta es:

$$G_p(s) = \frac{R_{2\_1}R_{2\_2}}{R_{1\_1}R_{1\_2}} \frac{1}{(R_{2\_\_1}C_{\_1}s+1)(R_{2\_2}C_{\_2}s+1)}$$

I.2.2. Red de adelanto discreta

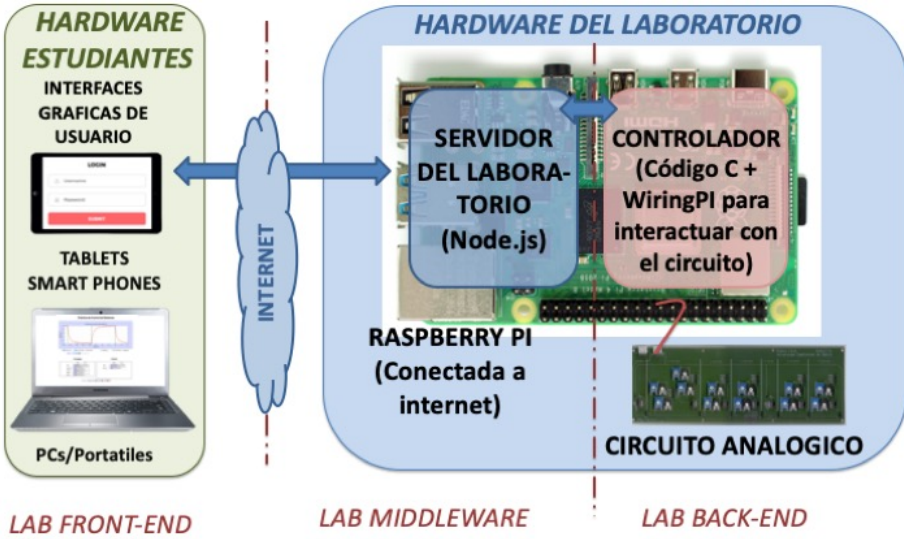
En esta sección se explica como discretizar la red de adelanto e introducir los parámetros obtenidos a través de la interfaz web de la práctica. Para detalles adicionales del funcionamiento en remoto del laboratorio, el alumno puede referirse a la documentación de apéndice C.

Implementación de un controlador discreto sobre el circuito analógico

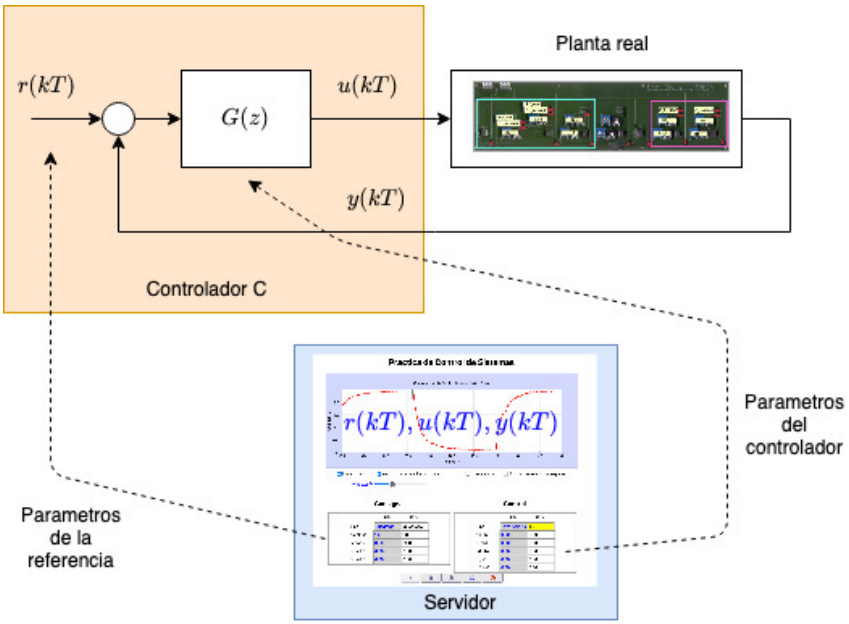
Para que los alumnos puedan probar de forma remota sus controladores discreto sobre la planta, esta se encuentra conectada a una Raspberry PI, sobre la que se ejecuta 1) el lazo de control correspondiente a una función de transferencia discreta de segundo orden y 2) un servidor web que permite que los alumnos accedan a los parámetros de la señal de referencia y del controlador, y observen la evolución de diferentes señales desde sus casa. La figura I.1 esquematiza los elementos hadware y software involucrados en el proceso y la relación que se establece, desde el punto de vista del Control de Sistemas, entre la planta (circuito analógico reconfigurable) y el controlador implementado.

El código implementado en la Raspberry PI y el interfaz gráfico del servidor permiten configurar (a través de los parámetros a_1, a_2, b_0, b_1 y b_2 de la interfaz gráfica de la práctica que se muestra en la Figura I.2) y ejecutar el comportamiento del controlador que mide la señal de salida, calcula la referencia y el error, computa la señal de control que genera la siguiente G(z) de segundo orden, y la aplica sobre la planta. Además, el interfaz gráfico también permite modificar la señal de entrada y sus parámetros, y por lo tanto ver la respuesta del sistema en lazo cerrado a una onda cuadrada.

$$G(z) = \frac{U(z)}{E(z)} = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2} \tag{I.1}$$



(a) Relación entre los elementos hardware y software del laboratorio



(b) Relación entre los elementos desde el punto de vista de control discreto

Figura I.1: Relación entre los elementos del laboratorio remoto cuando se aplica el controlador discreto

Discretización de la red e identificación de los parámetros de la G(z)

La función de transferencia continua del controlador red+ganancia obtenido en esta práctica es:

$$G(s) = K\frac{1 + aT_r s}{1 + T_r s} \tag{I.2}$$

Figura I.2: Interfaz Gráfica de la experiencia

Para discretizarla, utilizaremos la orden c2d, un periodo de muestreo de 0.1 segundos y el método por defecto (zoh) y bilineal (tustin), obteniéndose en ambos casos un controlador discreto de la forma:

$$G(z) = \frac{n_0 z + n_1}{z + d_1} \tag{I.3}$$

Para poder implementarlo como un controlador de orden 2 en vez de orden 1, únicamente hay que multiplicar el numerador y el denominador por z, y relacionar los parámetros del controlador discretizado I.4 con los de la función de transferencia de segundo orden discreta recogida en la Ecuación (I.1). Además, hay que introducirlos en el controlador discreto (IIR) a través de la pagina web que está conectada a la planta sobre la que hay que aplicar la red de adelanto.

$$G(z) = \frac{n_0 z^2 + n_1 z}{z^2 + d_1 z} \tag{I.4}$$

,

Es decir una vez dentro de la página web http://XXX.XX.XX.XXXX/ hay que elegir el controlador IIR, introducir los valores $b_0 = n_0$, $b_1 = n_1$, $b_2 = 0$, $a_1 = d_1$, $a_2 = 0$, elegir una señal de referencia (consigna) cuadrada de duración adecuada y amplitud 1, pulsar el botón update, y darle al play. A continuación, después de hacer doble-click sobre la imagen de la gráfica de evolución, observará la respuesta del sistema en lazo cerrado con el controlador elegido. Finalmente, el alumno

se desconectará de la práctica, para que se almacenen los datos en el fichero correspondiente y volverá a acceder a ella para poder descargarlos y representarlos con Matlab. La gráfica obtenida se comparará con la que se obtiene de forma simulada (con T=0.1s), a través de la superposición de ambas.

El alumno documentará los resultados obtenidos con el controlador discretizado con el método básico (zoh) y con el método bilineal (tustin).

Implementación digital de un controlador discreto

Finalmente, cabe indicar que el controlador discreto de segundo orden se implementa en el código C de la Raspberry PI, convirtiendo la $G(z)$ de la ecuación (I.1) en su correspondiente ecuación en diferencias,

$$u[kT] + a_1 u[(k-1)T] + a_2 u[(k-2)T] = b_0 e[kT] + b_1 e[(k-1)T] + b_2 e[(k-2)T] \tag{I.5}$$

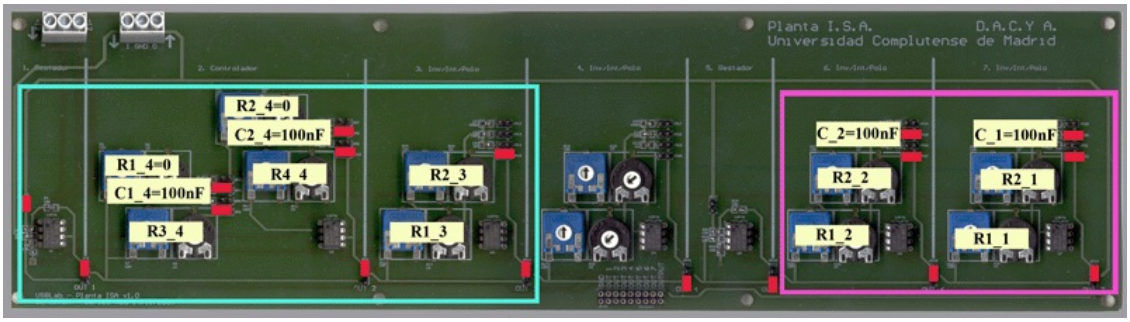
La implementación del controlador discreto de segundo orden se hará entonces, programando dentro de un bucle que se ejecuta en cada instante $t = nT$, las siguientes ecuaciones:

$$\begin{aligned}
e[kT] &= r[kT] - y[kT] \\
val &= -a_1 u[(k-1)T] - a_2 u[(k-2)T] + b_0 v[kT] + b_1 e[(k-1)T] + b_2 v[(k-2)T] \\
e[(k-2)T] &= e[(k-1)T] \\
e[(k-1)T] &= e[kT] \\
u[(k-2)T] &= u[(k-1)T] \\
u[(k-1)T] &= u[kT] \\
u[kT] &= val
\end{aligned} \tag{I.6}$$

donde $r[kT]$ es la referencia (consigna) que le entra al lazo de control, $y(kT)$ la respuesta de la planta, $e(kT)$ la señal de error que le entra al controlador discreto, y $u(kT)$ la señal de control que genera el controlador y que se le aplica a la planta en el instante kT.

I.2.3. Red de adelanto continua

El controlador implementado con el circuito analógico enmarcado en azul en la siguiente figura constituye una red de adelanto con una ganancia elegible, y un restador que realimenta la salida del circuito a la entrada del mismo.



Para analizar su comportamiento, estudiaremos el comportamiento de cada componente por separado.

El controlador esta configurado como una red de adelanto o retraso, y su función de transferencia viene dada por la siguiente expresión:

$$G(s) = -\frac{R_{4\_4}}{R_{3\_4}} \frac{(R_{3\_4}C_{1\_4}s+1)}{(R_{4\_4}C_{2\_4}s+1)}$$

El tener una ganancia fija e invertir la señal de entrada impide que por si mismo se pueda usar como un controlador cuya función de transferencia sea la ganancia que corrige el error estacionario y la red de adelanto diseñada. Para poder elegir la ganancia del controlador y corregir la inversión introducida por el circuito, se incluye el elemento inversor, cuya función de transferencia viene dada por:

$$G(s) = -\frac{R_{2\_3}}{R_{1\_3}}$$

Por medio del uso de estos dos elementos en serie se consigue tener un controlador cuya función de transferencia es:

$$G_c(s) = \frac{R_{2\_3}R_{4\_4}}{R_{1\_3}R_{3\_4}} \frac{(R_{3\_4}C_{1\_4}s+1)}{(R_{4\_4}C_{2\_4}s+1)}$$

Si se compara dicha expresión con la de una red de adelanto genérica con ganancia arbitraria K

$$G_c(s) = K \cdot \frac{aTs+1}{Ts+1}$$

se puede encontrar las relaciones entre los valores de K, a y T y los valores de los elementos (resistencias y condensadores) del controlador continuo implementado en el circuito.

Por último, para introducir estos elementos en el circuito, y cerrar el lazo de control es necesario: 1) modificar la posición de los jumpers relacionados con la etapa de control, y 2) conectar la salida del circuito a la entrada del restador que se encuentra a la entrada del circuito.

La configuración final del circuito completo, en lazo cerrado con el controlador diseñado, es la que se encuentra en el circuito de la figura de esta sección.

Para generar la señal que se aplica al sistema completo (planta+controlador+realimentación) y observar la respuesta del sistema en lazo cerrado, se utiliza, una vez sintonizados los valores de las resistencias y los condensadores en el laboratorio, la página web que da acceso remoto al circuito. En este caso, en el que el lazo de control ha sido cerrado sobre el circuito, hay que elegir la configuración de lazo abierto del controlador, que hace que este únicamente genere una referencia, la aplique sobre el circuito externo, y observe la respuesta. La forma de proceder en este caso, que se encuentra esquematizada en la Figura I.3, utiliza el programa de control únicamente para generar la señal de entrada y recoger la salida del circuito continuo reconfigurable.

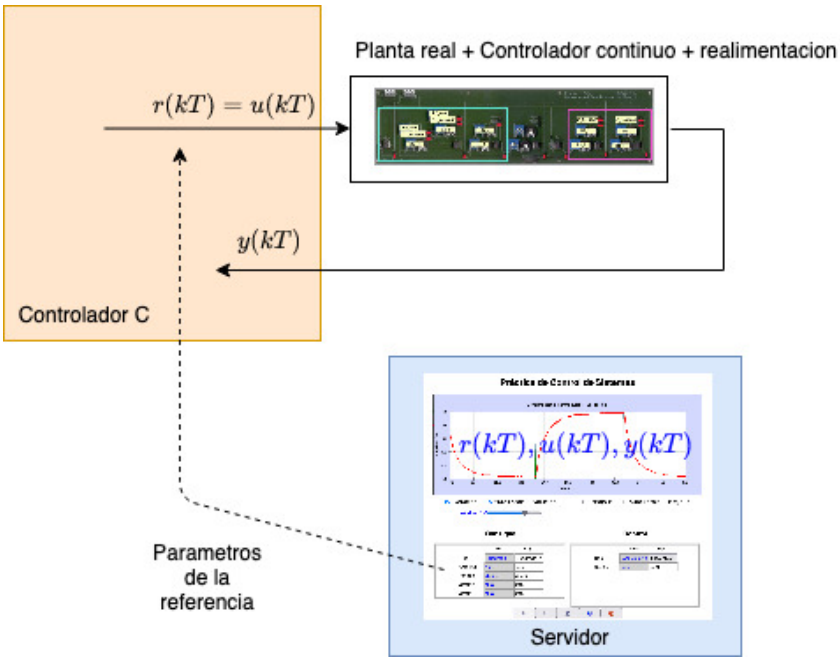


Figura I.3: Relación entre los elementos desde el punto de vista del controlador continuo

Apéndice J

Práctica de Control mediante reguladores PID

Esta práctica se lleva realizando, de forma presencial, en el laboratorio de Control de Sistemas de diferentes titulaciones desde el curso 2007/2008.

Sin embargo, durante el curso 2020-2021, debido a la situación semi-presencial de los laboratorios de la asignatura de Control de Sistemas de la titulación de Ingeniería Electrónica en Comunicaciones y a la disponibilidad de las herramientas desarrolladas en este PIMCD, ha sido mayoritariamente realizada de forma remota, ya que los alumnos han diseñado los controladores PID y probado sus versiones discretas desde sus casas, y asistido únicamente a una breve sesión del laboratorio presencial para probar el funcionamiento de las variantes continuas.

En las siguientes secciones, se muestra la documentación de la práctica que se les proporciona a los alumnos, actualizada con las modificaciones necesarias para reflejar la forma en la que han realizado la práctica durante este curso académico.

J.1. Guión de la práctica

A continuación se presentan los objetivos, la descripción y las tareas en las que se estructura la práctica de diseño e implementación de controladores PID, aplicados sobre el circuito analógico reconfigurable.

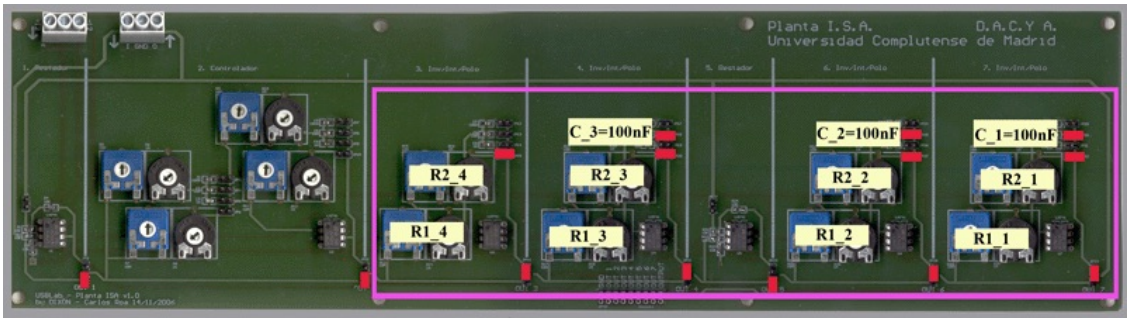
J.1.1. Objetivos

Los objetivos de esta práctica son:

- Diseño y sintonía de controladores de la familia PID mediante el método de máxima sensibilidad de Ziegler y Nichols.
- Optimización de los parámetros de los controladores obtenidos.
- Discretización de los controladores continuos y su implementación utilizando un ordenador.
- Implementación de los controladores continuos mediante un circuito analógico.
- Análisis y corrección del efecto del wind-up.

J.1.2. Descripción de la práctica

El sistema que se desea controlar en está práctica es el enmarcado en fuxia en la siguiente figura, y esta compuesto por tres polos reales en el semiplano negativo y una ganancia, colocados todos ellos en serie.



Un sistema de este tipo se puede controlar mediante un PID utilizando el segundo método (de máxima sensibilidad) propuesto por Ziegler y Nichols (resumido en la sección J.2.1 de la práctica). El objetivo de la práctica es diseñar y sintonizar dicho controlador y observar la respuesta del sistema controlado en lazo cerrado.

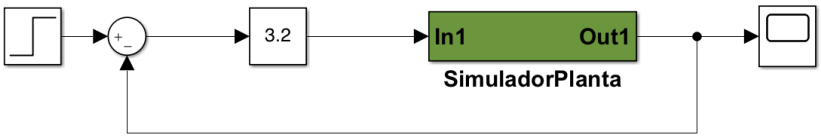
Tanto para el diseño del controlador como para una primera implementación de la acción de control y análisis de la respuesta del sistema en lazo cerrado se utilizará el módulo de Simulink proporcionado en el guión de la práctica. De esta forma el alumno podrá analizar, mediante simulación, el comportamiento de un sistema (del que desconoce la función de transferencia y que ya incluye las saturaciones de $\pm 12V$ propios de los amplificadores operacionales), el comportamiento del sistema regulado con las diferentes variantes de los controladores PID continuos, y el comportamiento en lazo cerrado de sus variantes discretas.

Una vez probado el comportamiento del sistema en simulación, los controladores continuos y discretos diseñados se probarán sobre el circuito real, utilizando para este fin el procedimiento que se detallará a lo largo del guión de esta práctica.

J.1.3. Realización de la práctica

La práctica se puede estructurar en las siguientes etapas:

1. Utilizando Simulink, realimentar el bloque Simulink que se proporciona con este guión (y que simula el comportamiento del sistema real) y controlarlo en serie con la planta con un controlador proporcionar. Ajustar la ganancia del controlador, hasta lograr aquella que permite obtener la K_{CR} y la T_{CR} del controlador PID (de acuerdo con el método descrito en la sección J.2.1). La amplitud de entrada escalón debe ser la adecuada para evitar que la señal de control a la entrada del bloque de la planta se salga del rango válido ($\pm 12V$).



2. Con los valores obtenidos para K_{CR} y T_{CR}, obtener las constantes de un controlador P, PI y PID.

3. Sustituir en el modelo Simulink la ganancia del primer paso por la versión discreta de los controladores diseñados y observar la respuesta del sistema en lazo cerrado utilizando Simulink sobre la PlantaSimulada.

4. A continuación el alumno realizará una de las dos tareas siguientes:

   - Optimizar los parámetros del controlador PID, intentando minimizar el tiempo de asentamiento y limitar la sobre-elongación a las dos terceras partes del valor que se obtiene con los parámetros del PID del método de Ziegler - Nichols.
   - Elegir un valor de la señal de entrada que haga visible el fenómeno del wind-up, e implementar en Simulink el método de seguimiento del punto de operación, eligiendo un valor del parámetro correspondiente adecuado, para corregirlo.

5. Discretizar los tres controladores propuestos (P, PI, PID) mediante un ZOH con el periodo de 0.01 segundos, y comprobará su funcionamiento sobre la planta simulada.

6. Calcular los valores de las resistencias y condensadores requeridos para implementar los tres controladores continuos diseñados. Utilizar el procedimiento descrito en la sección J.2.4.

7. Comprobar, de forma remota, el funcionamiento de los tres controladores discretizados sobre la planta real, utilizando el procedimiento que se describe en la sección J.2.5.

8. Comprobar, durante una sesión presencial del laboratorio, el funcionamiento de los tres controladores continuos sobre la planta real, siguiendo el procedimiento que se presenta en la sección J.2.6.

Finalmente, indicar que todas la etapas salvo la última pueden ser realizadas desde los equipos de los alumnos, sin que estos tengan que estar de forma presente en el laboratorio. Esto se debe a que para las seis primeras tareas el alumno utilizará Matlab y Simulink (herramientas de las que disponen los alumnos debido al acuerdo entre la UCM y Matlab) y que para la sexta se ha dado acceso a la experiencia a través de internet. Por lo tanto, únicamente deben realizar la tarea 8 de forma presencial, ya que en ella deben ajustar/sintonizar los pontenciómetros del Controlador con los valores que han calculado previamente.

J.2. Información adicional sobre la práctica

A continuación presentamos diferente información de utilidad para profundizar en diferentes aspectos y formas de proceder a lo largo de esta práctica.

J.2.1. Sintonía experimental de PIDs con el método Ziegler-Nichols de máxima verosimilitud

Ziegler-Nichols sugieren dos procedimientos experimentales para sintonizar controladores PID basados en:

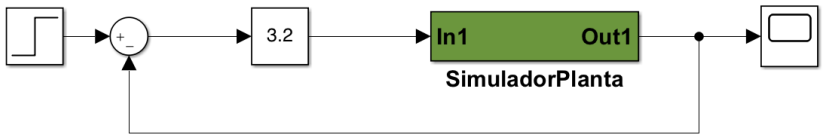
1. Analizar la respuesta experimental de una planta ante la entrada escalón.

2. Realimentar unitariamente el sistema y controlarlo en serie mediante una ganancia K que produzca la estabilidad marginal del sistema en lazo cerrado. Con el método se pretende obtener una sobreelongación máxima del 25 % en la respuesta a la entrada escalón.

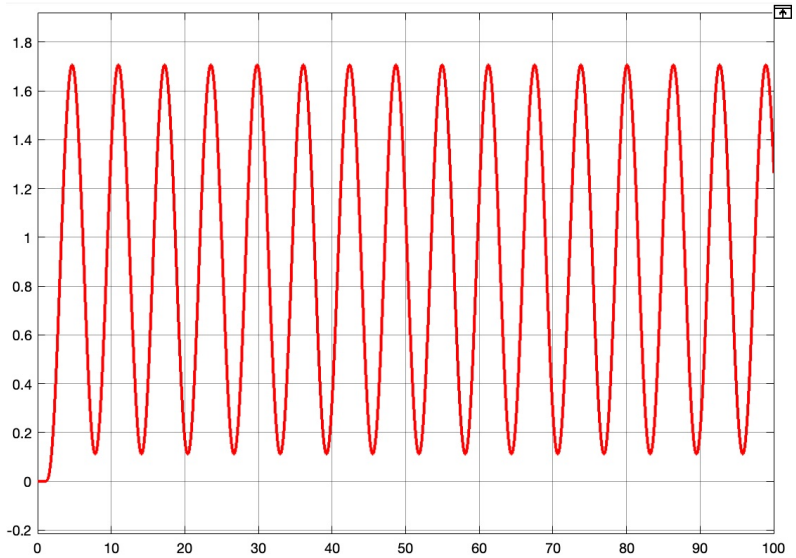
Estas reglas son empíricas y muy convenientes cuando se desconoce el modelo matemático de la planta, como ocurre en esta práctica. Para utilizar el segundo método (denominado habitualmente de máxima verosimilitud) en esta práctica, se realizan los siguientes pasos:

1. **Controlar la planta, realimentada unitariamente, mediante un controlador proporcional, determinar el valor de la K que hace que la salida del sistema sea una sinusoide mantenida (en el límite de la estabilidad), y calcular el periodo de oscilación T_C de dicha respuesta sinusoidal**.

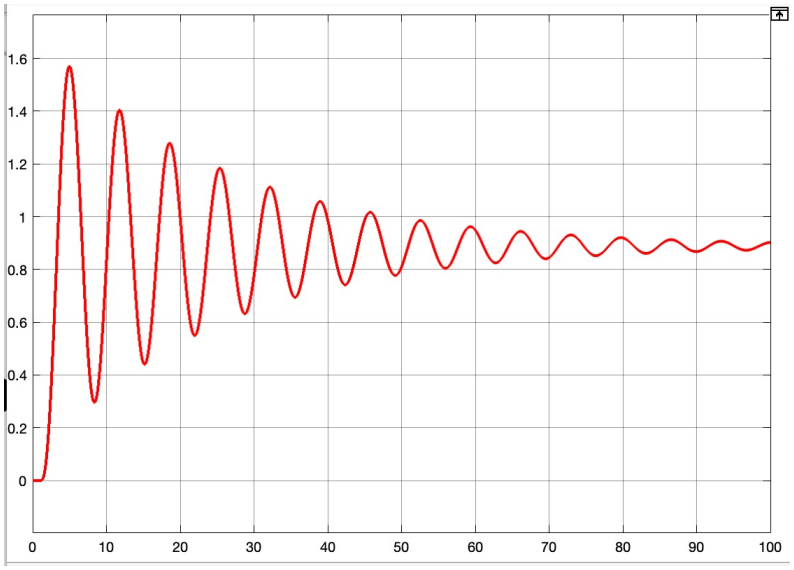
   Para realizar esta etapa con el modelo proporcionado construiremos el siguiente Modelo Simulink, en el que la constante proporcional se ha implementado mediante un slider para permitir que se ajuste su valor mientras se está ejecutando Simulink.



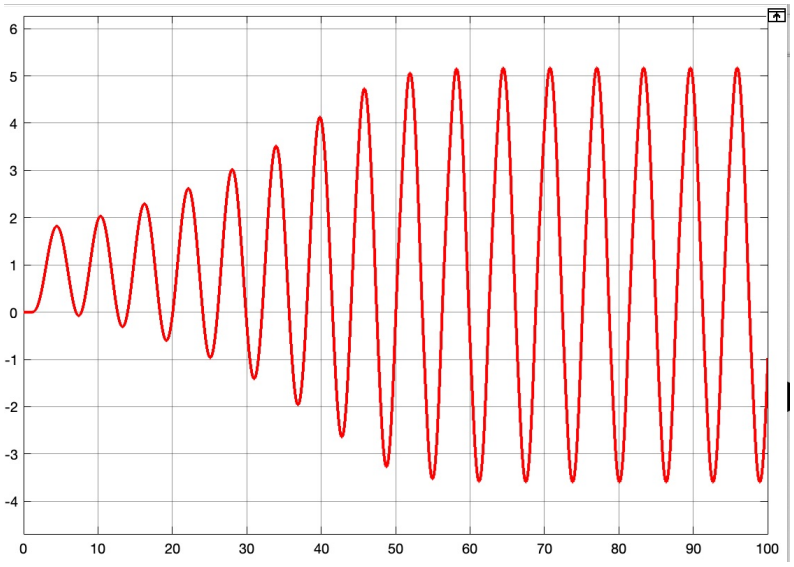
   A continuación, modificaremos el valor de la K hasta alcanzar el comportamiento de una onda sinusoidal como la de la siguiente figura y apuntaremos el valor de la K con la que se alcanza dicho comportamiento y el periodo de oscilación de la respuesta (medido, por ejemplo, entre dos máximos de la onda sinusoidal en el estacionario).



   Para determinar correctamente el valor de la K es conveniente empezar con un valor lo suficientemente bajo para que en la respuesta se observe el comportamiento de la siguiente figura (que corresponde al caso en el que la K hace que el lazo cerrado sea estable), hasta que alcancemos el comportamiento de la figura anterior.



Finalmente, y en lo referente a este paso, es importante indicar que el comportamiento de la siguiente figura se corresponde al caso inestable, aunque no se observe que la respuesta tiende, de forma oscilatoria, a infinito. El motivo por el que está queda confinada entre dos valores es la presencia de las saturaciones de los amplificadores operacionales que tiene la planta. Sin embargo, se puede observar que estamos en un comportamiento oscilatorio creciente en las primeras ondas de la figura, y por lo tanto, distinguirlo del comportamiento decreciente (figura anterior) y mantenido (de la figura en la que se muestra el limite de la estabilidad).



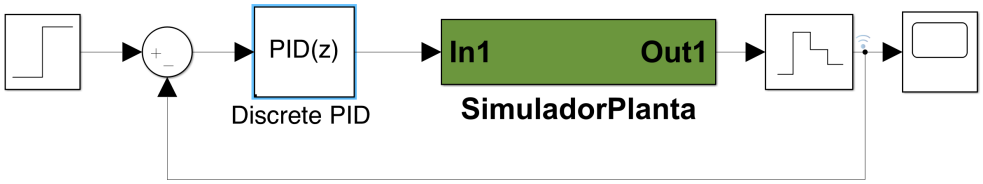
2. Obtener los valores de la constantes K_p, K_i y K_d de los controladores P, PI y PID, utilizando la siguiente tabla, y las relaciones K_i=K_p/T_i y $K_d = K_p \cdot T_d$.

	K_p	T_i	T_d
P	$0{,}5K$		
PI	$0{,}45K$	$0{,}8T_C$	
PID	$0{,}6K$	$0{,}5T_C$	$0{,}125T_C$

El alumno, a la hora de documentar los resultados de la practica, recogerá los valores calculados en una tabla similar a la anterior.

3. Comprobar y comparar el funcionamiento de cada uno de las variantes discretas de los tres controladores diseñados sobre el bloque PlantaSimulada.

   Para ello, basta con sustituir la ganancia slider del modelo Simulink que hemos utilizado en la primera etapa por un bloque PID de Matlab, e ir configurando las opciones para cada uno de los tres controladores.



   Además, en la prueba sobre la PlantaSimulada es posible replicar, sobre el mismo modelo Simulink, el lazo cerrado tres veces (para cada una con una variante del controlador PID), y recoger sobre la misma gráfica (un único bloque Scope de Simulink) el resultado de las simulaciones con los tres controladores.

J.2.2. Optimización de los parámetros del PID

Para optimizar los parámetros del PID, el alumno utilizará un modelo Simulink en el que los parámetros del PID sean variables que se puedan modificar desde las funciones de Matlab que desarrolle para implementar la función objetivo o de restricciones correspondientes. Estas funciones ejecutarán dicho modelo Simulink y evaluarán las funciones objetivo y restricciones correspondientes. Para evaluar los valores de dichas funciones se puede utilizar la orden de Matlab `stepinfo` sobre el resultado (vector de tiempos y de salida) de las simulación. Además, el alumno elegirá el método de optimización que considere más adecuado para sintonizar los parámetros del controlador.

J.2.3. Implementación del método de anti-windup

Como el controlador PID implementado por Simulink no incorpora la posibilidad de habilitar la acción de anti-windup, el alumno implementará todos los bloques del controlador continuo dentro de Simulink e incorporará el método de seguimiento del punto de operación visto en clase.

A continuación, mostrará las diferencias que existen entre el caso en el que se activa el anti-windup y en los casos en los que no. Para ello, será necesario que configure la entrada de la planta con los parámetros adecuados para que aparezca el fenómeno, y que, por lo tanto, pueda ser corregido con la activación del anti-windup.

J.2.4. Configuración de las resistencias y condensadores de los PIDs continuos

La realimentación unitaria y los controladores P, PI y PID pueden ser implementados sobre el circuito analógico configurable, mediante el uso de los jumpers adecuados y la sintonización de

las resistencias correspondientes. La información detallada del proceso se encuentra documentada en el apéndice D. Sin embargo, es conveniente hacer las siguientes puntualizaciones sobre cada controlador:

- **Controlador P**. La función de transferencia del controlador P que se puede seleccionar dentro del bloque de Control del circuito reconfigurable es una ganancia negativa ($G_c(s) = -\frac{R_4}{R_3}$). Como se necesitaría disponer de una ganancia positiva ($G_c(s) = K_p$), se desconectará, de los cuatro módulos en serie que constituyen la planta, el más cercano al controlador, que ha sido implementado como una ganancia unitaria negativa, y se ajustarán los valores de las resistencias del controlador de forma que $\frac{R_4}{R_3} = K_p$.

- **Controlador PI**. La función de transferencia del controlador PI que se puede seleccionar dentro del bloque de Control del circuito reconfigurable es una ganancia e integral negativa ($G_c(s) = -\frac{R_2C_2s+1}{R_3C_2s}$). Como la función de transferencia necesaria es positiva ($G_c(s) = K_p + \frac{K_i}{s} = \frac{K_ps+K_i}{s}$), se desconectará, de los cuatro módulos en serie que constituyen la planta, el más cercano al controlador (que ha sido implementado como una ganancia unitaria negativa), y se ajustarán los valores de R_2, R_3 y C_2 de forma que el controlador (sin el signo negativo) tome los valores deseados.

- **Controlador PID**. La función de transferencia del controlador PID que se puede seleccionar dentro del bloque de Control del circuito reconfigurable tiene la siguiente formula:

  $$G_c(s) = -\frac{(R_3C_1s+1)(R_2C_2s+1)}{R_3C_2s}$$

  La función de transferencia del PID es:

  $$G_c(s) = K_p + \frac{K_i}{s} + K_ds = \frac{K_ds^2 + K_ps + K_i}{s}$$

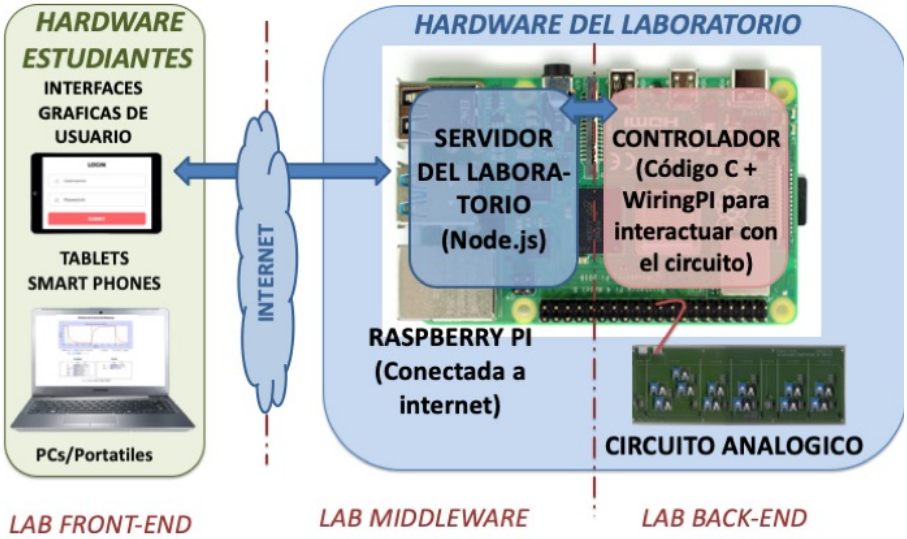
  A parte de la diferencia de signo (que se corrige, al igual que en los casos anteriores, deshabilitando el elemento de la planta que ha sido configurado con una ganancia negativa), puede ser necesario un parámetro adicional para fijar el tercer parámetro del controlador. Si fuese ese el caso, se puede añadir una ganancia K adicional al controlador, es decir hacer que

  $$G_c(s) = -K\frac{(R_3C_1s+1)(R_2C_2s+1)}{R_3C_2s}$$

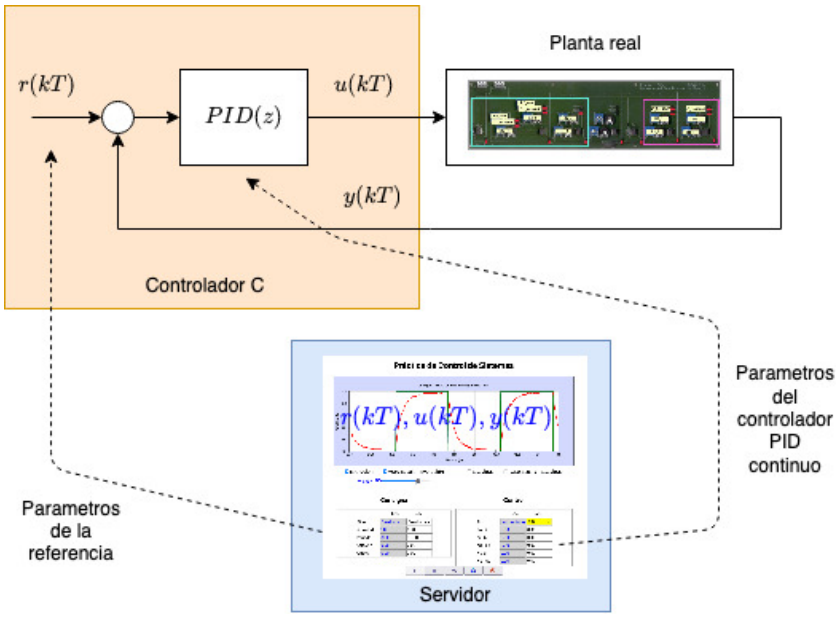
  mediante la modificación de la ganancia de uno de polos simples de la planta (tal y como se indicará en el laboratorio). Al desconocer la configuración exacta de la planta, el alumno deberá determinar los valores de R_3, C_1, R_2, C_2 y K para implementar el PID que tiene sintonizado.

J.2.5. Implementación de un PID discreto sobre el circuito analógico

Para que los alumnos puedan probar de forma remota sus PIDs discreto sobre la planta, esta se encuentra conectada a una Raspberry PI, sobre la que se ejecuta 1) el lazo de control correspondiente a un PID discreto configurable y 2) un servidor web que permite que los alumnos accedan a los parámetros de la señal de referencia y del controlador, y observen la evolución de diferentes señales desde sus casa. La figura J.1 esquematiza los elementos hardware y software involucrados en el proceso, y la relación que se establece, desde el punto de vista del Control de Sistemas, entre la planta (circuito analógico reconfigurable) y el controlador PID implementado.



(a) Relación entre los elementos hardware y software del laboratorio



(b) Relación entre los elementos desde el punto de vista de PID discreto

Figura J.1: Relación entre los elementos del laboratorio remoto cuando se aplica el PID discreto

El código implementado en la Raspberry PI y el interfaz gráfico del servidor permiten configurar (a través de los parámetros del controlador continuo K_p, K_i, K_d, N y NS de la interfaz gráfica de la practica que se muestra en la Figura J.2) y ejecutar el procedimiento que mide la señal de salida, calcula la referencia y el error, computa la señal de control que genera el PID discreto, y la aplica sobre la planta. Es importante recalcar que en la interfaz gráfica de la experiencia se introducen los parámetros del PID continuo y el Código en C que cierra el lazo de control sobre

Figura J.2: Interfaz Gráfica de la experiencia

la planta real, calcula a partir de ellos los valores necesarios para implementar el PID en discreto. Finalmente, conviene indicar que el entorno gráfico también permite modificar la señal de entrada y sus parámetros, y por lo tanto ver la respuesta del sistema en lazo cerrado a una onda cuadrada.

J.2.6. Implementación del PID continuo sobre la planta analógica

Para analizar el comportamiento de las diferentes variantes del PID continuo sobre el circuito analógico es necesario configurar el módulo de Control de dicho circuito con los valores de las resistencias calculados en las sección J.2.4 y conectar los jumpers correspondientes a cada variante del PID y al sumador/restador que compara la salida de la planta con la señal de referencia, y que por lo tanto, es el responsable de introducir la realimentación correspondiente en el sistema.

Para generar la señal que se aplica al sistema completo (planta+controlador+realimentación) y observar la respuesta del sistema en lazo cerrado, se utiliza, una vez sintonizados los valores de las resistencias y los condensadores en el laboratorio, la página web que da acceso remoto al circuito. En este caso, en el que el lazo de control ha sido cerrado sobre el circuito, hay que elegir la configuración de lazo abierto del controlador, que hace que este únicamente genere una referencia, la aplique sobre el circuito externo, y observe la respuesta. La forma de proceder en este caso, que se encuentra esquematizada en la Figura J.3, utiliza el programa de control únicamente para generar la señal de entrada y recoger la salida del circuito continuo reconfigurable.

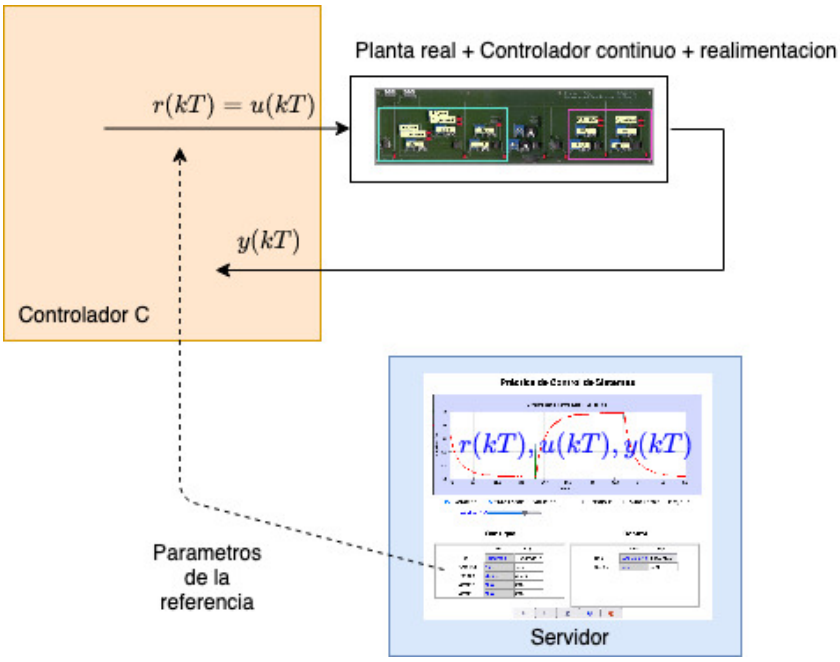


Figura J.3: Relación entre los elementos desde el punto de vista del PID continuo

Bibliografía

[1] E. Besada-Portas, "EJS-TwinCAT 3.0: Nuevo laboratorio remoto integrado en moodle para prácticas de control de ingeniería de sistemas y automática," Proyecto de Innovación Educativa y Mejora de la Calidad Docente (PIMCD), Universidad Complutense de Madrid, 2014.

[2] E. Besada-Portas, J. A. Lopez-Orozco, L. de la Torre, and J. M. de la Cruz, "Remote control laboratory using EJS applets and TwinCAT programmable logic controllers," *IEEE Transactions on Education*, vol. 56, no. 2, pp. 156–164, 2013.

[3] F. Esquembre, "Facilitating the creation of virtual and remote laboratories for science and engineering education," in *IFAC Workshop on Internet Based Control Education*, 2015.

[4] J. Bermudez-Ortega, E. Besada-Portas, J. A. Lopez-Orozco, J. Chacon, and J. M. de la Cruz, "Developing web TwinCAT PLC-based remote control laboratories for modern web-browsers or mobile devices," in *2016 IEEE Conference on Control Applications (CCA)*, pp. 810–815, 2016.

[5] J. Bermudez-Ortega, E. Besada-Portas, J. A. Lopez-Orozco, J. A. Bonache-Seco, and J. M. de la Cruz, "Remote web-based control laboratory for mobile devices based on EJsS, Raspberry Pi and Node.js," in *IFAC Workshop on Internet Based Control Education (IBCE)*, pp. 158–163, 2015.

[6] J. Bermudez-Ortega, E. Besada-Portas, L. de la Torre, J. A. Lopez-Orozco, and J. M. de la Cruz, "Lightweight node.js ejss-based web server for remote control laboratories," in *IFAC Symposium on Advances in Control Education (ACE)*, pp. 127–132, 2016.

[7] J. Sáenz, J. Chacón, L. De La Torre, A. Visioli, and S. Dormido, "Open and low-cost virtual and remote labs on control engineering," *IEEE Access*, vol. 3, pp. 805–814, 2015.

[8] J. Chacon, J. Saenz, L. D. l. Torre, J. M. Diaz, and F. Esquembre, "Design of a low-cost air levitation system for teaching control engineering," *Sensors*, vol. 17, no. 10, 2017.

[9] J. Chacon, E. Besada-Portas, G. Carazo-Barbero, and J. A. Lopez-Orozco, "Enhancing EJsS with extension plugins," *Electronics*, vol. 10, no. 3, 2021.

[10] I. Aizpuru-Rueda, E. Besada-Portas, J. Chacón, and J. Lopez-Orozco, "Despliegue automático de laboratorios remotos extendiendo las capacidades de ejss," in *XL Jordanas de Automática (JJAA)*, pp. 242–249, 2019.

[11] "Repositorio del PIMCD." `https://gitlab.com/jcsombria/pimcd-2019-2020`. Accessed: 2021-06-15.

[12] "Dobot." `https://www.dobot.cc`. Accessed: 2021-06-15.

[13] "Pydobot." `https://github.com/luismesas/pydobot`. Accessed: 2021-06-15.

[14] "Instalación de Raspbian." https://www.raspberrypi.org/documentation/installation/installing-images/. Accessed: 2021-06-15.

[15] "Activación del SSH en la Raspberry PI." https://www.raspberrypi.org/documentation/remote-access/ssh/. Accessed: 2021-06-15.

[16] "Repositorio del labratorio remoto." https://github.com/jcsombria/ReNoLabs. Accessed: 2021-06-15.

[17] "Instalación de ejss." https://gitlab.com/ejsS/JavaScriptEditor/release. Accessed: 2021-06-15.

[18] E. Besada-Portas, "Diseño de prácticas reales para un laboratorio unificado para las asignatuas del área de ISA," Proyecto de Innovación Educativa y Mejora de la Calidad Docente (PIMCD), Universidad Complutense de Madrid, 2006/07.

[19] Microchip, "MCP4802/4812/4822 8/10/12-Bit Dual Voltaje Output Digital-to Analog Converter with internal Vref and SPI Interface," 2015.

[20] Microchip, "MCP3004/3008 2.7V 4-Channel/8-Channel 10-Bit A/D Converterswith SPI Serial Interface," 2015.

[21] Motorola, "JFET Input Operational Amplifiers, TL081C,AC/TL082C,AC/TL84C,AC," 1997.